Archiving Strategies for On-line Decision Making in Evolutionary Multi-Objective Optimization

Mario Köppen, Kaori Yoshida, Masato Tsuru, Yuji Oie
Kyushu Institute of Technology
680-04 Kawazu, Iizuka, Fukuoka 820-8502, Japan
Email: mkoeppen@ieee.org, kaori@ai.kyutech.ac.jp, tsuru@ndrc.kyutech.ac.jp, oie@cse.kyutech.ac.jp

***Abstract*—In this paper, we are studying a generalized version of the Strength Pareto Evolutionary Algorithm 2 (SPEA2). By replacing the algorithmic-internal role of the Pareto-dominance relation with a different, not necessarily transitive relation, the algorithm can become capable to search for the maximum set of the replacing relation. Thus, the SPEA2 algorithm can also become capable of on-line decision making during the evolutionary search. The approach will be exemplified by using the maxmin-fair relation among objective vectors that avoids extremities of a subset of a few objectives only. But without the Pareto-dominance relation, also the way of filling the archive of the SPEA2 algorithm has to be revised. The matter becomes complicated by the fact that the replacing relations might be only transitive in some subpspaces - as it is the case for the maxmin-fair relation. Different methods for maintaining an archive are proposed and experimentally compared. The results of the study suggest, for the case of a larger number of objectives, to use an archiving strategy, which is based on transitive subspaces of the replacing relation.**

I. Introduction

Recently, the usage of evolutionary computation to search the Pareto set of a multi-objective optimization problem has gained a rapidly increasing interest (see [1] for a review of recent work). In contrary to single-objective optimization, there is often a trade-off between competing objectives, and the solution set usually contains more than one element. Often, reference is made to a decision maker as a means to incorporate further application knowledge, which is not fully represented by the various objectives, into the approach for soliciting final solutions from the obtained trade-off sets. In typical scenarios, this is done off-line, i.e. after the Pareto set of the search problem has been approximated, and by qualifying different parts of this Pareto set to establish an external decision.

In this paper, we want to continue with the rather infrequently considered approach to incorporate the decision maker already into the evolutionary multi-objective search, and are presenting a general framework for such on-line decision making. For introducing this approach, it is necessary to see what kind of generic external knowledge can there be in multi-objective optimization that cannot be simply represented by an additional objective. Actually, such knowledge sources can be easily identified, but they are often only given in human terms. Just to give a few examples: there might be the interest to have *flexible* solutions for a problem, i.e. solutions, which are not changing strongly under slight changes in the environment. In a modern human-centered application context, one might look for a *natural* solution to a problem, i.e. a solution that resembles the human way of handling similar problems and fulfills some kind of simplicity. Also, there might be the interest for *fair* solutions, i.e. solutions that avoid the trade-off extremes where only a few objectives attain their extreme values.

The formalization of such knowledge sources can be either a mathematical relation between two solutions, stating e.g. that solution A is more flexible/natural/fair than solution B, or by the definition of a kind of stable state, where e.g. a solution A is flexible/natural/fair since any attempt to change it into a solution B will be in opposition to some criterion derived from the knowledge source.

In case a formal representation of such a knowledge source as a relation between two solutions can be provided, it is possible to use such a relation in an evolutionary multi-objective optimization (EMO) algorithm, e.g. by a corresponding alteration of the selection operator. Then, as will be shown below, the whole algorithm can be opted to search for the maximum set of the relation instead of the Pareto front.

However, a main problem for the usage of alternate selection operators is that the underlying formal relations among solutions (which replace the Pareto-dominance relation) are not necessarily transitive anymore. Taking or examples from above again (flexibility, naturalness, fairness), there is no need to require that a formalization of such relations has to be transitive. Thus, the question of archiving, i.e. a means for maintaining a set of good solutions in the sense of the used formal relation becomes of high importance[1].

In this paper, we will provide a general framework for the on-line incorporation of a decision maker into the evolutionary search by using alternative selection operators, and consider different archiving strategies. The main focus will be on the provision of fairness among the objectives, and we can base

[1]There is an additional interest in such approaches that should be mentioned here. In case of a larger number of objectives (which should be understand as substantially more than three objectives, and is commonly referred to as many-objective optimization), the exponentially decreasing frequency of the appearance of the Pareto-dominance relation among i.i.d. random vectors gives a large hinderation to the application of EMOs in such a context. Using alternative selection operators, this frequency might fall less than exponential (and here we will study an example where this probability falls linearily with the number of objectives, for example). But also here, previously studied amendments to the Pareto-dominance relation are not necessarily transitive anymore

our study on a number of related works in the network congestion control.

Section II of this paper will present the general framework, and a subsection will exemplify this for the maxmin-fairness, which was already intensively studied in the context of congestion control for elastic networks. Then, section III will discuss the possible archiving approaches, taking the non-transitivity of the derived maxmin-fair relation into account. Section IV will provide a comparative study of these archiving approaches, for gaining a better understanding of their advantages and drawbacks. A conclusion section is provided at the end of the paper.

II. A FRAMEWORK FOR ON-LINE DECISION MAKING IN EVOLUTIONARY SEARCH

For an arbitrary relation $\sim$ we have the notions of a *maximum set* and a *best set*[2]:

Definition 1. *For a feasible space X and a relation $\sim$ (seen as subset of $X \times X$), the maximum set X_M is the set of all elements $x \in X$, for which no other element $y \in X$ exists such that $y \sim x$.*

Definition 2. *For a feasible space X and a relation $\sim$, the best set is the subset of all elements $x \in X$ such that $x \sim y$ for any other $y \in X$ holds.*

For the following discussions, we also need the formal definition of the Pareto-dominance relation:

Definition 3. *A point $x \in R^n_+$ (where $R^n_+ = \{x \in R^n | x_i \geq 0, i = 1, 2, \ldots, n\}$) is said to Pareto-dominate a point $y \in R^n_+$, formally $x >_P y$, if for all $i = 1, \ldots, n$ $x_i \geq y_i$ and for at least one $j = 1, \ldots, n$ $x_j > y_j$.*

If $\sim$ would be an equivalence relation, the maximum set for example would be the set of all elements of the feasible space, which are different to any other element. However, these notions will rather target cases where the relation is somehow related to the concept of ranking, voting or deciding. For example, for the Pareto-dominance relation, the maximum set equals to what is commonly called the Pareto front or Pareto set, and the best set is usually empty. But here, we will not require any other "typical" property of order relations like completeness or transitivity.

While keeping this general picture, we observe that among many evolutionary multi-objective optimization algorithms, especially the Strength Pareto Evolutionary Algorithm 2 (SPEA2)[3], by simple modification, can be made capable *to find the maximum set of an arbitrary relation*. To see this, we shortly recall the processing of this algorithm. The SPEA2 algorithm is basically a standard evolutionary algorithm, with only the selection operator replaced. The processing of SPEA2 assigns a so-called S-value to each individual in the population. For each individual of the population at some generation, it is counted how often its objectives (seen as a point in the real-valued objective space) are Pareto-dominating other individuals of the population. This is the so-called Pareto-strengths of this individual. Then, the S-value of each individual is the sum of the Pareto-strengths of all individuals, by which it is dominated. Clearly, a solution is considered weaker if its S-value is larger, and all individuals having objectives on the Pareto-front of the population have the S-value 0. However, this notion of an S-value can be expanded to any other relation as well.

Thus, we propose a generic set-up for a search algorithm for general maximum sets. The search problem is formalized by a mapping from a *feature space* X into an *objective space* Y. Among the elements of Y, a relation $\simeq_R$ is considered. The search problem is to find feasible feature vectors that are mapped into the maximum set of all feasible objective vectors. Note that in such a context, the objective space Y is not necessarily numerical.

1. A first parent population of N individual points X_i is randomly initialized. For each point X_i, the corresponding objective Y_i is assigned. To each individual i, the number P_i of other individuals j such that $Y_i \simeq_R Y_j$ is counted. Then, each individual i gets the S-value assigned as $S_i = \sum_{j \neq i, y_j \simeq_R y_i} P_j$.
2. From the parent population of the former step, N children are generated by using selection, cross-over and mutation operators. The selection is tournament selection: two pairs of individuals are drawn at random, and the one with the lower S-value of each pair is kept. By uniform cross-over, a new individual is generated. With probability p_{mut}, each component of an individuals' position gets uniformly modified. However, other genetic operators can be applied as well.
3. The union of the former parent population and the new set of children is established, and the S-values are computed (note that the parents' S-values might change). The union set is sorted according to increasing S-values, and the first N individuals are taken as the new parent generation.
4. If some stopping criteria is met (like number of parent generations), stop, otherwise, go to step 2.

Thus, we are able to provide algorithms especially soliciting specific properties of the Pareto set, as long as we are able to provide a corresponding relation. But the SPEA2 algorithm also employs an archive that contains the Pareto set of all visited points in search space (with some additional means to reduce the size of this archive, if it grows too large, and ensuring diversity in the archive). Since we propose to replace the Pareto-dominance relation in SPEA2 by other relations, the question is what to do with the archive in this case. This will be the main topic of the following discussions. For doing so, we will at first provide a working version of the generic algorithm given above in the following subsection.

A. Instantiation by the Maxmin-fair relation

Recently, a special version of such a modified SPEA2 algorithm has been proposed for fair network resource allocation and congestion control[4], and some of the points made there have to be recalled here. In general, we are considering a fairness relation in the general domain of points x in R^n_+.

Then, among two points x and y, to be *maxmin-fair related* is defined as follows:

Definition 4. *A point $x \in R^n_+$ is said to be maxmin-fair related to a point $y \in R^n_+$, $x \simeq_F y$, if for any $y_i > x_i$ there exists at least one $j \neq i$ such that $x_j \leq x_i$ and $y_j < x_j$ ($i, j = 1, \ldots, n$).*

This definition is derived from the common definition of maxmin-fairness, where a point x is sought that is maxmin-fair related to any other point y of the feasible space [5]. However, the complication is there that in the general case, often no such single maxmin-fairness point exists, since this relation is not complete (see below).

The motivation for such a definition of fairness comes from the field of network resource allocation and congestion control, where global optimization may sometimes lead to unwanted solutions. For example, consider a so-called parking-slot scenario [6]. There, a sender 1 is sharing links in a network with two other senders 2 and 3. With sender 2, she has to share link 1, and with sender 3 link 2. Both links have a maximum capacity m for data transmission. If the incoming traffic for a link exceeds this maximum capacity (i.e. congestion occurs), the link distributes the overexcess traffic in a linear manner (the traffic of sender i is denoted by x_i): If $x_1 + x_2 > m$, then x_1 can only transmit $y_1 = m \star x_1/(x_1 + x_2)$, and similarly for senders 2 and 3. If we want to maximize the total throughput $T = y_1 + y_2 + y_3$ of this network, we have to consider whether links are getting congested or not. For simplicity, we will assume $x_2 = x_3$. Then, if both links are congested, the throughput T will be

$$T = m\frac{x_1}{x_1 + x_2} + 2m\frac{x_2}{x_1 + x_2} = m\left(1 + \frac{x_2}{x_1 + x2}\right) \quad (1)$$

and since all $x_i \geq 0$, this expression attains its maximum $2m$ if $x_1 = 0$. However, if at least one of the links is not congested, the total throughput will be surely lower than $2m$, and the global optimum for the total throughput in this network is indeed $2m$ for $x_1 = 0$ and $x_2 = x_3 \geq m$. In this optimum, sender 1 is not allowed to send any traffic into the network.

The maxmin-fairness approach, which is an attempt to avoid such a situation, is to give higher priority to the lower demands. Roughly spoken, a solution to resource allocation is considered to be maxmin-fair, if for any other solution, which gives more to someone, there is always someone else, who already receives the same or less and will get lesser in this other solution. In simple scenarios like the parking slots just studied, the maxmin-fair resource allocation exists and is unique. Also, there are algorithms like Progressive Filling of Bottleneck-Links [6] that allow for finding this maxmin-fair point.

Some notes on the definition of the maxmin-fair relation: if there is no component, at which x is smaller than y (i.e. at any component $i, x_i \geq y_i$), then the condition of the definition is also considered to be fulfilled. This also means:

Theorem 1. *In R^n, $x >_P y$ implies $x \simeq_F y$. The maximum set of the maxmin-fair relation is a subset of the maximum set of the Pareto-dominance relation.*

In the following, we provide some other mathematical properties of the maxmin-fair relation.

Theorem 2. *The maxmin-fair relation is reflexive.*

This follows directly from the definition.

Theorem 3. *For the feasible space R^n, the maxmin-fair relation is not complete.*

Providing a counter-example is sufficient: for $x = (6, 5, 4)$ and $y = (3, 5, 6)$, neither $x \simeq_F y$ nor $y \simeq_F x$ holds, since only their smallest components are getting larger in the corresponding other vector.

Theorem 4. *For the feasible space R^n, the maxmin-fair relation is not transitive.*

This can be seen from the counterexample $x = (4, 5), y = (3, 2)$ and $z = (5, 1)$. Here, $x \simeq_F y$ and $y \simeq_F z$, but not $x \simeq_F z$. Due to this lack of transitivity, we also did not use a symbol containing ">" in the notation.

Definition 5. *A set of points from R^n is said to be* all-different *if any pair of different components taken from the same or different points is different.*

Theorem 5. *Within a subset of all-different points, where each of them has its (unique) minimum component at the same position k, the maxmin-fair relation is transitive.*

Proof: This follows directly from the definition of the maxmin-fair relation. Assume two vectors a and b both having their smallest component at index k with $a_k > b_k$. Then, if already $a >_P b$, also $a \simeq_F b$ follows directly. If not, there must exist at least one other component $l \neq k$ with $b_l > a_l$ and we can see that the definition of maxmin-fair relation is fulfilled for vector a: for all such l, $b_l > a_l$ but there is always $a_k \leq a_l$ and $b_k < a_k$. So, $a \simeq_F b$, and in this subspace, the maxmin-fair relation equals the maximum relation for the k-th component. ∎

Finally, we recall an important property for the size of the maximum set of the maxmin-fair relation[4].

Theorem 6. *The maximum set of the maxmin-fair relation for any subset of R^n with all-different points can have at most n elements.*

Proof: This can be seen by assuming a set M of m points, which are pairwise not maxmin-fair related to each other. Any maximum set of a maxmin-fair relation for some feasible space will have this form, since in the maximum set, no other point of the feasible space, including the maximum set itself, can be maxmin-fair related to any other point of the maximum set. With x^*_i we denote the point x that has a minimal value among all points in M in regard with component i. Having all-different points ensures that for each i, there is exactly one such point in M. If some point x, which takes its minimum at component i, is not maxmin-fair related to another point y, then $x_i < y_i$ has to hold. Thus, if x is not maxmin-fair related to any different $y \in M$, then it has to be $x = x_i^*$ and therefore there could be at most n points such that each of which is not maxmin-fair related to any other point. ■

This maximum size can also always be attained. For any n, there is also a set of n points such that none of them is maxmin-fair related to any other. For example, take as the i-th component of the k-th point the value $n((k+i-1) mod n) - (k-1)$.

III. Archiving strategies for non-transitive selection relations

EMO and also single-objective evolutionary optimization algorithms (EC) usually maintain an archive of the "best solutions" found over the concourse of the whole algorithm over several generations. For EC, this is a rather straightforward task, since any set of real or integer numbers has a unique supremum (or infimum). For EMO algorithms, this archive contains a sub-set of the Pareto-front of all solutions considered so far. Transitivity of the Pareto-dominance relation ensures the usability of the archive to provide the final outcome of an algorithm. Moreover, an EMO algorithm cannot be considered to be complete without a specific archiving strategy, which also gives the means for any performance evaluation. However, if considering the maxmin-fair relation, it is generally not transitive, and the question comes up how to register "good solutions" during the evolutionary search here.

To answer this question, we recall two of the mathematical properties of this relation that can help to establish a sufficient archiving strategy. At first, it has to be remembered that an element of the maximum set is also an element of the Pareto set. So, an archive could be simply based on the Pareto-dominance relation in a similar manner as for most other EMO algorithms. But also, transitivity in subspaces can be employed for archiving. Essentially the fact that the maxmin-fair relation is transitive among all all-different vectors v_i, where element k is the smallest component can be employed for deriving a suitable archiving strategy.

Based on this consideration of the mathematical properties of the maxmin-fair relation, we can consider three paradigms for an archiving strategy: basing the archive update on the Pareto-dominance relation, on the maxmin-fair relation, or on transitive subspaces. These archiving strategies will be detailed in the following subsections. In each version, the archive A gets updated after each generation. The objective vector of each element of the population of size N is used to update the archive.

A. Maxmin-fair relation based archive

The first approach is to use the same relation for the archiving as for the selection operator, despite of the lack of full transitivity (or subspace transitivity only).

1) Initially, $A = \emptyset$.
2) After each generation, and for each v_i being the objective vector of individual $i = 1, \ldots, N$, and for each $a \in A$: if there is no $a \in A$ such that $a \simeq_F v_i$, then $A = A \cup \{v_i\}$.
3) If $v_i \in A$, then all $a \in A$ are removed, for which v_i is maxmin-fair related to a: $A = A \setminus \{a \mid v_i \simeq_F a\}$.

For reasons given before, the archive will never contain more elements than the number of objectives (or components of v_i).

B. Pareto-dominance relation based archive

Alternatively, one can left the task of promoting maxmin-fairness to the selection operation, and use the Pareto-dominance relation for archiving. The procedure is similar to the former one, with the role of maxmin-fair relation replaced by the Pareto-dominance relation. But additionally, the growing of the archive size has to be limited. Especially in case of a larger number of objectives, the Pareto-dominance relation becomes rapidly unlikely, and thus more and more elements get added to the archive.

1) Initially, $A = \emptyset$.
2) After each generation, and for each v_i being the objective vector of individual $i = 1, \ldots, N$, and for each $a \in A$: if there is no $a \in A$ such that $a >_P v_i$, then $A = A \cup \{v_i\}$.
3) If $v_i \in A$, then all $a \in A$ are removed that are Pareto dominated by v_i: $A = A \setminus \{a \mid v_i >_P a\}$.
4) If $|A| > \alpha$, where α is the maximum allowed size for the archive, then all elements of A are sorted by their sum of components, and A is replaced by the first α elements of the sorted set.

C. Transitive subspace based archive

Here, the archive is specially designed from the specific mathematical properties of the selection operation. In our case, we have seen that the maxmin-fair relation is transitive in some subspaces. Subspace i equals the open subset of all objective vectors, where component i is the smallest. This can be used to define another archiving strategy:

1) Initially, A is a set of size n (the number of objectives) with each element being nil.
2) After each generation, and for each v_i being the objective vector of individual $i = 1, \ldots, N$, and for each $a \in A$: determine the set of indices $I = \{i_1, \ldots, i_K\}$ such that for each $i_k \in I$ $v_{i,i_k} = \min_i(v_i)$, i.e. which are equal to the smallest element of v_i.
3) Now, for each $i_k \in I$: if A_{i_k} is (still) nil, then A_{i_k} is set to v_i. If there is already an entry A_{i_k} in the archive, and $v_{i,i_k} > A_{i_k,i_k}$, i.e the individuals' objective vector v_i has a larger smallest component at i_k then the vector already in the archive, the vector in the archive is replaced by v_i.

Also here, the archive size is at most the number of objectives.

IV. Results

It has to be taken into account that the archiving strategy is not there to improve the algorithm alone. Its main purpose is to solicit solution tested over the generations, which are in favour of the optimization goal.

TABLE I
RESULTS FOR THE RELAXED BOTTLENECKS PROBLEM FOR THE THREE DIFFERENT ARCHIVING STRATEGIES (SEE TEXT FOR DETAILS).

method	$\|A\|$	$y_2(lastgen)$	$y_3(lastgen)$	$y_2(bestgen)$	$y_3(bestgen)$	#hits
5 objectives						
fairdom	2	146.97	99.26	147.37	99.45	90%
pareto	234.5	152.09	102.12	152.91	103.49	100%
subspace	5	137.66	96.88	139.30	96.88	80%
10 objectives						
fairdom	6	92.35	72.55	112.51	71.45	0%
pareto	308.9	107.96	88.22	117.76	89.61	10%
subspace	10	93.85	83.41	99.69	84.83	10%
15 objectives						
fairdom	6.9	73.68	74.10	80.77	75.45	0%
pareto	287	79.57	90.15	89.79	89.37	10%
subspace	15	88.03	85.71	91.19	88.12	20%

We have studied the algorithm presented in section II for a scalable network congestion problem with known solution. We consider the "parking slot" scenario, where the network is a linear chain of at least four links and each traffic enters at one node and leaves at the next. Only one traffic enters the network at the first node and leaves at the last one. The maximum capacity of each link but two is set to 100 (whatever unit), and each link maintains linear congestion control. For link 2 and 3, we used 200 and 150 as maximum capacity.

If all links would have equal maximum capacity of 100, the maxmin-fair point is achieved by sending an amount of 50 for each sender into the network. That this is a maxmin-fair point can be easily seen: if any sender entering at node i and leaving at node $i+1$ would increase its traffic to more than 50, the traffic of the sender entering at node 1 and leaving at the last node would be reduced to less than 50. So, there is a sender who already receives less or equal and who would receive less in case of traffic increase, and the definition for maxmin-fairness is fulfilled. The same can be seen for the first sender.

If there is a higher capacity provided for some links, then for each such link there is exactly one sender sharing this link with the sender entering at node 1 and leaving at the last node. The sender on node 1 is restricted to 50 by the fairness demands to other links of capacity 100, so the other sender can take full benefit of the increased capacity, without affecting any other traffic share. So, in the example case studied here, providing capacities 200 and 150 to links 2 and 3 allows sender 2 and 3 to send traffic amounts 150 and 100 into the network, and 50 for all others. For reasons just given we want to call such kind of problems "relaxed bottlenecks problem."

For all these studies, the algorithm settings were as follows: population size $m = 20$, 300 generations, mutation probability $p_{mut} = 0.3$ and all average results were taken over 10 runs of the algorithm. At the beginning, each component of each individual of the population was randomly initialized with an i.i.d drawn number from $[0, 100]$. For the *pareto* method, the maximum allowed archive size was set to 500. The settings for population size, number of generations and maximum archive size were appropriate to handle these dimensions, but it could not be directly expanded to a larger dimension. For e.g. 20 objectives and after 300 generations, there wouldn't be enough points close enough to 50 to allow for a further evaluation.

The results can be found in table I. For three choices of the problem dimension (5, 10 and 15), all three archiving strategies were tested. The first entry is the average size $|A|$ of the archive after 300 generations. The next two entries in each row give the 2nd and 3rd component of that element of the archive in the last generation, which has the largest sum of these two components. In the same manner, the next two entries show the values maximizing the sum of the 2nd and 3rd component over *all* generations. The last entry is the number of runs, where this element was within 10% of the maxmin-fairness point.

While there was no obvious problem to allocate a traffic of about 50 to any other sender than 2 and 3 (which was already reported in [4]), the relaxed bottlenecks, as can be seen, cause some problems to the algorithm, especially when the number of senders is increasing. The search effort is strongly increasing with the number of objectives, and the best results yielded after 300 generations differ more and more from the possible best values 150 and 100.

While all three methods do not behave extremely different, some tendencies can be clearly seen. In general, the method *pareto* provides the best results, followed by *fairdom* and last seems *subspace*. However, for the many-objective case with 15 senders, the method *subspace* becomes comparable in performance with the *pareto* method. Given the more simple processing and the fact that the subspace method was more often able to come closer to the true optimum, this method seems to be more attractive for the many-objective case.

There is a rather simple explanation for the lower performance of archiving by transitive subspaces in the case of a lower number of objectives: the lack of updates for the subspaces for the second and third component. Since the goal is to increase these two values to values larger than 50, they will never provide the minimum component for better solutions. So, the archive entries here get stuck after the first generations. In case of e.g. five objectives, this means that nearly half of the archive members are not being updated anymore, and the archiving strategy is less efficient. For more objectives, this effect can be more and more neglected.

The results for all methods also indicate that the best results were not always found in the last generation. This means, even in the case of the *pareto* method, that the archiving strategy may always loose good solutions again due to intransitivity, and that a kind of meta-archiving (as it was done here for example by monitoring the archive over all generations) seems nevertheless unavoidable.

V. Conclusions

In this paper, a generic framework for the evolutionary finding of maximum sets of arbitrary relations (preferrably relations related to ranking, sorting, deciding or voting) was presented. The framework is based on a corresponding modification of the SPEA2 algorithm by replacing the Pareto-dominance relation in the internal computation of the S-values with the other relation. This approach could be exemplified by using the maxmin-fair relation, which represents maxmin-fairness and is motivated from related research in network resource allocation and congestion control: a point of a feasible space is said to be maxmin fair if an increase in any of its components is accompanied by a decrease of another component of the same point, which is already smaller or equal. This relation has a maximum set, which is a subset of the Pareto set and has at most as much components as there are objectives. However, depending on the feasible space, this relation is not always transitive. Thus, while the internal processing of SPEA2 can be easily modified to use the maxmin-fair relation instead of the Pareto-dominance relation, the usage of the same relation for the update of the archive of SPEA2 is questionable. So, other strategies for archiving have been comparatively studied in this paper as well. Essentially there are three different choices for an archiving strategy: using the replacing relation like the maxmin-fair relation; using the Pareto-dominance relation; or specifying an archive due to the subspaces, where the relation is transitive. By a comparative study, the approach using the Pareto-dominance relation seems to show slightly better general performance. However, for a larger number of objectives, the transitive subspace approach shows a comparable performance. Due to other factors like lower computational effort, this seems to be the recommendable choice for an archiving strategy.

Acknowledgment

This work was partially supported by the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (S) (No. 18100001)

References

[1] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*, 2nd ed. Springer Berlin, Heidelberg, 2007.

[2] A. Sen, *Collective Choice and Social Welfare*. Holden-Day, 1970.

[3] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," in *EUROGEN 2001, Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, Eds., Athens, Greece, 2002, pp. 95–100.

[4] M. Köppen, M. Tsuru, and Y. Oie, "Evolutionary approach to maxmin-fair network-resource allocation," in *Proceedings of The 2008 International Symposium on Applications and the Internet (SAINT 2008)*, vol. SAINT08 Workshops, 2008.

[5] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

[6] J. Y. L. Boudec. Rate adaptation, congestion control and fairness: a tutorial. [Online]. Available: http://ica1www.epfl.ch/PS_files/LEB3132.pdf