Comparison of Evolutionary Multi-Objective Optimization Algorithms for the Utilization of Fairness in Network Control

Mario Köppen, Rodrigo Verschae, Kaori Yoshida, Masato Tsuru
Kyushu Institute of Technology
Network Design and Research Center (NDRC)
680-4 Kawazu, Iizuka, Fukuoka, Japan
mkoeppen@ieee.org

***Abstract*—We use design principles of evolutionary multi-objective optimization algorithms to define algorithms capable of approximating maximum sets of relations in general. The specific case of fairness relations is considered here, which play a prominent role in the control of resource sharing in data networks. We study maxmin fairness allocation in networks with linear congestion control. Among various design principles, the concepts behind Strength Pareto Evolutionary Algorithm, and the Multi-Objective Particle Swarm Optimization achieve comparable best performance (with the used parameterization within 10% of the fairness state components for up to 20 objectives).**

***Index Terms*—evolutionary computation, meta-heuristics, multi-objective optimization, fairness, maxmin fairness, general fairness relation, Pareto dominance**

I. Introduction

Over the past decade, meta-heuristic approaches to multi-objective optimization, also referred to as Evolutionary Multi-Objective Optimization (EMO) algorithms, have gained increasing attention [1]. For the extension of single objective optimization, where the search is for an extreme value of an objective function within a feasible space, the comparison of individuals is now based on the Pareto-dominance relation, and the goal becomes to approximate the maximum set of this relation (called Pareto front for the induced relation in the objective space). Thus, and in addition to the expansion of the concept of an extreme of a set of real values to the case of more than one dimension, it is also the first time that meta-heuristic algorithms were designed in order to approximate *maximum sets* of general relations. If taking the common definition of a relation as subset of the direct product of sets (set with itself in this case), the maximum set is defined as the set of elements, to which no other element is in that relation. The definition is efficient in the sense that each relation has a uniquely specified maximum set (but it might be empty, for example for the relation $\{(a,b),(b,c),(c,a)\}$ of the set (a,b,c)). The task to approximate maximum sets can be also seen as generalized optimization.

With the present study, we want to investigate the use of meta-heuristics for a new class of relations. Recently, in network engineering, studies are stronger focusing on the achievement of fairness in sharing resources (for example congestion control, wireless channel allocation) [2]. The reason for this is that it was learned that even in simple scenarios it can happen that global optimization will give inacceptable results, where users are virtually excluded from the use of any resource [3]. This line of research has lead to a formalization of fairness, which can be also based on a relation. Thus, the goal here is to consider the potential of meta-heuristic algorithms that are capable of approaching the maximum set of such fairness relations. For the design of such algorithms, EMO algorithms already provide a rich resource. The main focus of this paper is to compare the more prominent design principles of EMO algorithms with regard to fairness relations. For the purpose of evaluation, we selected a problem that initially led to the notion of fairness, the traffic congestion problem [4]. In case of a specific fairness (maxmin fairness), the fair state of traffic for such networks can be directly computed, and this allows for a justification of the performance of different designs for fairness meta-heuristic algorithms.

The paper is structured according to this goal. Section II collects the main facts about fairness, presents the extension to a general fairness relation, and also lists the design principles for multi-objective search that are used in this paper. Section III than introduces the performed experiments, presents the results and is concluded by a discussion. The paper ends with a conclusion section.

II. Fairness and Meta-Heuristics

In this section, we will describe the various means to define fairness, its generalization, and present several ways how meta-heuristic algorithms can be designed in order to search for the maximum set of fairness relations. For doing the latter, we will be guided by examples for searching the maximum set of the Pareto-dominance relation by means of evolutionary multi-objective optimization algorithms.

A. Formal Approaches to Fairness

The common-sense concept of *fairness* or its opposite *unfairness* is often used in our daily communication, for example when discussing sport contests or the effect of new political measures on some companies. Notably less effort has been done on the formalization of fairness. Especially in computer science, we may find related considerations in processor scheduling, while accessing a common resource like a memory bank. Here, the concept of *n-fairness* was introduced, where one processor allows the foregoing processor in the schedule to access the memory bank at least n times [5]. This can be used as a boundary condition for the formalization of scheduling problems as combinatorial optimization problems. Such approaches require a *fairness indicator*, i.e. a numerical procedure leading to a real value, the increase of which (or keeping within bounds) is a secondary objective of the optimization. In case of network engineering, the most common approach for an indicator is the Jain fairness indicator [6]. Given a share $x = \{x_i\}$ of $i = 1, \cdots, n$ resources (the limitations of which are established by additional formalizations), this indicator is defined as

$$J_x = \frac{(\sum_{i=1}^{n} x_i)^2}{n \cdot \sum_{i=1}^{n} x_i^2} \tag{1}$$

In practice, these shared resources are related to user traffic in networks under congestion situation (where not all users can fully realize their traffic demands) or wireless channel allocation of base stations. However, the use of such indicators is implicitly based on the assumption that there is an complete and transitive ordering relation among the elements of the feasible set of shares: for having an incomplete relation, the corresponding indicator values would still allow for a comparison of any pair of incomparable shares, and for having a non-transitive relation, also the indicator values would still imply transitivity for any triple of shares violating transitivity. Making such an assumption in advance is rather surprising, as most ordering relations in multimodal domain are known to be incomplete (for example the Pareto dominance relation, see next section). Therefore, in more general attempts to formalize fairness, we also find fairness related to equilibrium: among the feasible set of shares, one is considered to be fair if any attempt to increase its components (within the feasible space) would cause a reduction for components that are already smaller. This is most directly reflected by the so-called *maxmin fairness* [4], [2].

Definition 1. *An element x of the feasible space is* maxmin fair *if for any other element y and for each component y_i of y, which is larger than the corresponding component x_i of x there is a $j \neq i$ for which x_j is (already) smaller than or equal to x_i and y_j is smaller than x_j. In other words, the "price" for the improvement of one component of a maxmin fair point is always paid by another component, which already got less of a share.*

The interest of such a definition is to solicit a single element of the feasible space. However, it is not guaranteed that such an element will exist at all, thus rendering the definition somehow inefficient. In order to change this to an efficient definition, we have proposed to establish a fairness relation instead, and then to look for the maximum set of this relation (i.e. the set of shares, for which no other share is in such a fairness relation) [7]. Thus one is roughly following the manner, in which Pareto equilibrium is expanded to the Pareto dominance relation. As a consequence, the maximum set of the relation may contain more than one element, and it would need an additional *decision maker* to select one share from this maximum set. Formally, the maxmin fairness relation then is defined as:

Definition 2. *An element x of the feasible space is* maxmin fair dominating *an element y of same space, if for each component y_i of y, which is larger than the corresponding component x_i of x there is a $j \neq i$ for which x_j is (already) smaller than or equal to x_i and y_j is smaller than x_j.*

In other words, as a relation we are comparing two vectors (shares) x and y with regard to components of y that have become larger than the corresponding component of x and thus might indicate some improvement. However, for any such improvement, it might be that other components will become smaller. If among these components, who get smaller, is at least one that was already smaller than or is equal to the improved component before its change, then we will not consider this a fair improvement. Fairness dominance relation reflects the fact that this "unfairness" happens at least once and vector x cannot be fair(ly) improved to the other vector y.

The maxmin fairness comes out to be rather restrictive. One point is: if the minimum component of a vector x is already smaller than the corresponding component of y, and the minimum value appears only once among the components, then it cannot maxmin fair dominate the vector y since there is no other equal-or-smaller component of x and thus no "candidate" for unfair improvement[1]. The other point is: the relation does not take the magnitude of the order relations into account. This can become a problem in a situation, where small fluctuations of components occur (including the modifications of components appearing in meta-heuristic algorithms that we are going to discuss below).

For this reason, the so-called *proportional fairness* [8] has become more popular. It is departing from the maxmin fairness approach of comparing individual components by comparing the net effect of all improvements versus the net effect of all reductions, and requires the former to be larger than the latter. With an additional normalization, the definition (expressed in terms of a relation) is:

Definition 3. *An element x of the feasible space is considered* proportional fair dominating *an element y of the feasible space iff*

$$\sum_{i=1,\ldots,n} \frac{y_i - x_i}{x_i} \leq 0 \tag{2}$$

holds.

For better reflecting utility, it has been sometimes considered as special case of α-fairness:

[1]This also guides to the fact that the maximum set of this relation is composed of the vectors having the minimum components for at least one dimension.

Definition 4. *An element x of feasible space is considered α* fair dominating *an element y of the feasible space iff*

$$\sum_{i=1,\ldots,n} \frac{y_i - x_i}{x_i^{\alpha}} \leq 0 \quad (3)$$

holds.

This α fairness is also nearly always converging to *maxmin fairness* if the exponent α goes to infinity (nearly always, because it will not converge for a set of measure 0). But there are also other fairness relations: in lexicographic maxmin fairness dominance, which is based on lexmin lexicographical ordering of vectors, a vector x is dominating a vector y, if its minimum component is larger [9]. In case both are equal, the second smallest components are compared etc.

Recently, a generalization of such fairness relations has been proposed [10]. It needs a sheaf of functions $\{P(c)\}$, which are parameterized by a scalar value c, and whose functions for different value of the parameter do not cross each other. Each of the functions are everywhere non-convex, and converges to the value 0 as one of its components goes to infinity (in case of two dimensions, a typical example for such a sheaf of functions can be generated from the coincidence $P(x, y) = xy = c$). For a given point (x_0, y_0) with positive components, we define a fairness relation in the following way (and which can be directly expanded to the case of more than two dimension):

1) Consider the curve $P(x, y) = P(x_0, y_0)$ and its tangent at the point (x_0, y_0). We call it *iso-average*, and P the generating averaging operator.
2) The intersection of the feasible space and the half-plane below the tangent is the subspace dominated by (x_0, y_0). It is said that (x_0, y_0) is *more fair* than any other solution (x, y) within this subspace.

For example, if we choose $P(x, y) = xy$, then this procedure leads to the proportional fair dominance. If we choose other power means (with negative exponent), we get α-fair dominance, and if $P(x, y)$ converges this way to the $\min(x, y)$, the corresponding fairness relations nearly always converge to maxmin fairness.

The advantage of this generalization, besides of allowing to define an uncountable set of fairness relations, incorporating utility etc. is also that common properties of all former fairness relations can be studied at once.

Obviously, such fairness relations are not complete. But in addition, they are in general not transitive. Both can be seen from the example shown in fig. 1. For a point (x_0, y_0), the function selected from the sheaf is $P(x, y) = P(x_0, y_0)$. All points in the shaded area, i.e. the area bounded by the positive coordinate axis and the area below the tangent on the curve P at the point (x_0, y_0) are fair dominated by that point. For such a dominated point (x_1, y_1) the corresponding tangent is also shown (the corresponding sheaf function $P(x, y) = P(x_1, y_1)$ is not plotted), and thus, the point (x_2, y_2) is fair dominated by (x_1, y_1) but not fair dominated by (x_0, y_0). So, this is a case where a point a fair dominates a point b, and b dominates another point c, but where a is not dominating c. However,

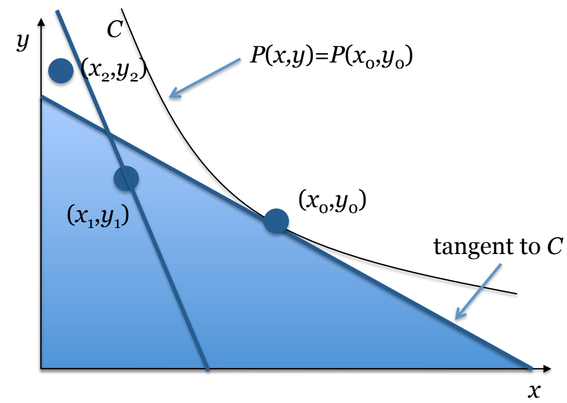

Fig. 1. Example for a case where a generalized fairness relation is not transitive.

despite non-transitivity in such cases, c fair dominating a will never happen. Thus, the intransitivity here is essentially incomplete, but never conflicting.

Despite of lacking transitivity (and thus not being an order relation), the definition of fair dominance is efficient in the sense that it uniquely specifies a maximum set. Also, without giving the necessary analytic details here for space reason, an analysis of the relation gives that for convex feasible spaces the maximum set will contain at least one element (exactly one except under very specific circumstances).

B. Design Principles of Multi-Objective Meta-Heuristics

For coming up with algorithms to search for the maximum set of a (generalized) fair dominance relation, we will refer to the well-known case of Pareto dominance relation. To recall Pareto-dominance, we provide the definition shortly.

Definition 5. *A point $x \in R_+^n$ 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 $j = 1, \ldots, n$ $x_j > y_j$.*

The Pareto-dominance relation is not complete (simple example are the two points $(1, 2)$ and $(2, 1)$, where no points dominates the other), but it is transitive. Pareto-dominance also has relations to the fair dominance relations introduced in the former subsection. To name the most essential one: Pareto dominance implies fair dominance. Another quick glance on fig. 1 reveals that the area, which is Pareto-dominated by a point (x_0, y_0) (it is the rectangle with diagonal corner points (x_0, y_0) and $(0, 0)$) is always contained in the area below the tangent. Thus, the maximum set of the fair dominance relation will be always a subset of the maximum set of the Pareto-dominance relation (if not, for some point x of the maximum set of the fair dominance relation, there would be a point y Pareto-dominating x, but then, y would also fair dominate x and x cannot belong to the maximum set). An important difference is that Pareto-dominance is not mutually relating components with each other, while fair dominance does. Thus, the scope for the practical application of fair dominance relations is restricted to cases, where the "meaning" of a component (e.g. its related objective) is not dependent on the index assigned to it. However, formal processing of a Pareto-dominance relation, if replaced by fair dominance relation, is not affected, as this processing will never refer to a particular index of a component.

We are interested in the means to approach the maximum set of the Pareto dominance relation (Pareto set from now on), and how they give an approach to handle the same task for the fair dominance relations. Over the past years, the field of Evolutionary Multi-Objective Optimization (EMO) has been rapidly expanding. Starting from the beginning of the 90s' with the proposal of a so-called VEGA algorithm (Vector Evaluated Genetic Algorithms [11]), it has been learned that there are a number of ways to adjust the standard evolutionary computation meta-heuristic to make algorithms capable of searching for Pareto sets. Next we will recall the most prominent ways of doing such an extension. This will give a cursor for the corresponding design of meta-heuristics for fair dominance relations.

In the following, the term Pareto set will refer to a maximum set within the feature space. In multi-objective optimization, elements x of the maximum set of the *induced* Pareto-dominance relation are searched. If $f(x)$ is the co-domain of the objective function, i.e. composed of points $f(x)$ in *objective space* of points x in *feature space*, x dominates another feature point y iff $f(x)$ Pareto dominates $f(y)$. The image of the Pareto set in objective space of the mapping f is also called Pareto front.

1) General modifications: In general, EMO algorithms follow a similar procedure to other evolutionary algorithms. A first generation population of individuals is initialized (randomly, or by incorporating domain knowledge), and then iteratively one generation is derived from the foregoing one (the parent generation): children are generated by selecting pairs of parents according to their "strong" fitness, cross-over is performed, then mutation. Possibly other genetic operators intervene in this processing. From the combined children and parents, the next generation is selected, often by the same means as selection candidates for crossover. The iteration repeats until a stopping criterion is fulfilled (like maximum number of generations). Then, the population, or its archive (see below) are evaluated to get the result of the algorithm.

The need for a modification of a standard evolutionary computation algorithm, in order to also handle multiple objectives (or vector-valued fitness functions) comes from the fact that vectors can become incomparable, and that a set of vectors can have more than one "maximum." In order to apply genetic operators like cross-over, it needs a way of selecting individuals according to some "strength" in the same manner as in the case of a single objective. All other processing of genetic operators (like standard mutation) is not affected by the circumstance that there is more than one objective. Thus, it essentially needs to design corresponding means for *selection* among individuals. Four of them will be recalled in the next subsections.

The other modification is the means for deciding on a result of the run of such an algorithm. While in single objective case, a set will always comprise a single maximum value, in case of multiple objectives this will not always happen. The common means to allocate a result to an EMO is the use of an archive: during the processing of the EMO, newly generated individuals (i.e. points of feasible space) are always tested against the elements already within the archive. If any point in the archive does not dominate the point, it will be added to the archive. After adding, all points now dominated by the new member will be removed from the archive. The result of the EMO than can be taken from the archive, or from the final population. It should be noted that the use of an archive always refers to transitivity of Pareto-dominance relation: once a member is added to the archive, and which is dominating another member, by removing the latter one it is nevertheless ensured that the new member will also dominate all visited points ever dominated by the removed member.

Besides of these two aspects, it should be noted that also the comparison of algorithms is affected from the change to multiple objectives, but this is out of the scope of present study.

2) SPEA: In the Strength Pareto Evolutionary Algorithm (version 2, SPEA2 [12]), the basic concept for providing a modified selection is to assign a "Pareto strength," or S-value to each member of the population. At first, for each individual i of current population, the number R_i of other individuals dominated by individual i is counted. In the second step, for each individual i the sum of all R_j values for individuals j dominating individual i is computed as S_i. Then, the S_i values are used like a fitness in single objective case for the other genetic operators.

It can be immediately seen that this way of computing S-values is not depending on the used relation. The only thing guaranteed is that elements of the maximum set of a relation will receive the S-value 0. Thus, its use for fair dominance relation is straightforward, and a corresponding algorithm has already been explored on basic problems in [7].

3) NSGA: By far the most popular EMO these days is NSGA-II (Non-dominated Sorting Genetic Algorithm) [13]. Here, introducing a dual measure used for comparing vectors modifies the selection. At first, a rank is assigned to each individual of the population. Individuals currently in the Pareto set get rank 0 assigned. Then, starting with $n = 0$, iteratively all individuals of rank n are removed, and the rank $(n + 1)$ is assigned to all individuals of the maximum set of the remaining set. This repeats until no more individuals without rank are available. For selection, individuals are first compared by their rank. But if the ranks are equal, a secondary measure is used, based on some other needs of the algorithm. In case of original NSGA-II, this is the so-called crowding distance. Its value is smaller the more the individual is residing in a part of the feasible space, where already other individuals of current population are located.

Also here, the concept of ranking can be directly employed for fair dominance relations, as it is only referring to the maximum set of the relation. However, a corresponding complexity reduction introduced with NSGA-II for the rank computation is based on transitivity of the Pareto dominance relation, and this cannot be performed in case of fair dominance relations.

The crowding distance is a plain computation from the vector components, without reference to a ranking relation.

4) PAES: The Pareto Archived Evolution Strategy (PAES [14]) is not based on a population. A single element (the *parent*) is kept, and variations of it are produced, called *candidates*. If the candidate Pareto-dominates the parent, it replaces the parent. If it is dominated, it is disposed. If there is no dominance relation between parent and candidate, a more complex comparison procedure is initiated, making use of the archive, which is maintained in the same manner as in any other EMO (see above). Basically a candidate replaces a parent also when it succeeds to be added to the archive, and is disposed if it is dominated by any member of the archive. If it is neither dominating nor dominated by the archive, additional means (similar to NSGA-II) related to crowding are taken into account: if the candidate resides in a more crowded part of the feasible space (with respect to the members of the archive) than the parent, it is also disposed, otherwise it replaces the parent.

PAES makes only use of the dominance relation. The role of the archive has been described as giving "guidance" to the search [14], and so the incomplete transitivity of fair dominance relation still allows for specifying a version of the algorithm by replacing Pareto dominance with fair dominance in all stages of the archive processing.

5) MOPSO: Particle Swarm Optimization (PSO), while not a genuine evolutionary computation, has also been modified into a multi-objective version in an increasing number of ways (so-called MOPSOs). The survey presented in [15] gives an early overview over a number of algorithms, but also points out the general principles behind all these extensions.

We will not repeat the PSO equations in detail here (see [16] for details), as we are only interested in the extension of a MOPSO to the case of fair dominance relation. In PSO, a swarm of particles is moving in feasible space. Each particle maintains a (local) best position of its own trajectory, and a global best position from the neighborhood of the particle (having a corresponding topology defined in advance). In each generation, positions of all particles are updated by a velocity term, which is the weighted sum of three components: the inertia term equal to former velocity; distance to local best, multiplied by a random number; and distance to global best, multiplied by a random number. For a multi-objective version, the ways of updating local and global best have to be modified, as simple numerical comparison of the fitness values is no longer possible. The update of the local best is typically based on the Pareto-dominance relation. Only if the particle after position update (and usually also a mutation) is dominating the former local best, it replaces the local best position. For global best selection, the algorithms usually maintain a set of mutually non-dominating positions, so-called *leaders*, and some means to select among the leaders and to update the leaders. In the simplest way, the leaders update is handled like an archive of an EMO (this concept is also employed here), and the leader selection is random.

For extension to fair dominance case, similar arguments as for PAES apply: we can handle the set of leaders like the PAES archive, base the local best update on the fair dominance relation, and do not need adjust any other part of the MOPSO algorithm.

C. Scaling of Meta-Heuristics for High Dimensionality

A last comment refers to the number of objectives. With respect to objective space dimension, i.e. the number of objectives, EMO algorithms are (unfortunately) strongly affected by the "curse of dimensionality." One reason is that the Pareto-dominance becomes sparse, i.e. the number of cases, where in a high-dimensional space a random vector is Pareto-dominating another random vector falls exponentially with the number of objectives. For example, NSGA-II, usually demonstrating excellent performance in the handling of even highly multimodal problems with 2 or 3 objectives, can fail for the so-called *many-objective optimization* since internally all the individuals of the population get rank 0 assigned, and the comparison then is completely resting on the crowding distance. Another reason is the limited scope of random modifications to cover substantial parts of the objective space.

The term "many objectives" is not describing a crisp border with respect to the number of objectives, but it can be carefully said that usually 2-3 objectives are within the application domain of an EMO, while the performance starts to fall heavily (or the need for increasing the search efforts increases drastically) at least between 10 and 20 objectives.

In case of the target network design problems, it might be a little bit discouraging to study such meta-heuristic approaches, since in real-world networks, the number of objectives might refer to number of users, and this is far more than "10-20." But it also has to be taken into account that probably, no other algorithms for solving fairness problems are available, and that a control of a data network does not necessarily need to be global. Thus, it is also important to learn about the potential scope of the application of meta-heuristic algorithms, and to find the base point for further improvements by comparing algorithm design principles.

III. Comparison for Fair Network Congestion Control

For studying the algorithm design principles, we take the problem of network congestion control. There, a data network is assumed as a (bi-directional) graph $G(N, L)$ with nodes N and edges/links L, where to each link of L a capacity c is assigned. Traffic is a set of tuples (s, r, π, m), where node s is the sender, r the receiver, π is a path, i.e. a linked chain of links connecting sender and receiver, and m is the amount of traffic. While utilizing a network, various traffics may increasingly start to share links, thus exceeding the maximum capacity of such links. In this case, a *congestion control* becomes activated. Here, we employ the most common linear congestion control scheme, where the routers (hosted at the network nodes) reduce incoming traffics proportionally to their amounts, such that the sum becomes equal to the maximum capacity.

In this problem domain, maxmin fairness (it was initially proposed in this context) among the traffics can be computed by a so-called "bottleneck filling" algorithm. Here, for a set of traffic tuples, the traffic amount is kept as a variable and set according to the maxmin fairness state at the end of the algorithm.

Algorithm FSM: Fairness State for Traffic Set

---

1) Set the *remaining set* R to the traffic set. Set the traffic m for all traffics to 0.
2) While R is not empty, repeat the following steps:
3) For all links l_i used by the paths in R, get the number w_i of paths that pass through this link.
4) Find the minimum of $m_i = c_i/w_i$ where c_i is the capacity of link l_i.
5) Add m_i to the traffics for all senders through the links with minimal m_i.
6) Remove all elements of R with senders through links, for which m_i is minimal.
7) Set new capacities of network links $c_i \leftarrow c_i - m_i * w_i$.

Note that such an algorithm for general traffics is only known for the maxmin fairness so far. By using this algorithm and a selection of traffic sets, where congestion is enforced, we can directly justify the performance of the modified algorithms.

A. Material and Method

For testing the proposed fairness meta-heuristic algorithms (FMA), a number of aspects have to be taken into account, which will be listed in the following.

1) Traffic Generation: It has to be noted that for fairness in network routing, only the paths along with their traffic and link sharing are relevant. The role of the graphs is to grant feasibility of such traffics. The goal to generate random test traffic and to enforce congestion (note that normally the practical concern is to avoid congestion) was achieved by the following measures:

1) The number of nodes was taken rather small (here 6).
2) To all links, random weights were assigned, and then, the Floyd algorithm was used to find minimum weighted paths between any possible sender-receiver pair (the Floyd algorithm does this for all pairs at once).
3) From the result of the Floyd algorithm, the paths for a number of random sender-receiver pairs were selected (not taking care for multiple occurrence of senders or receivers).
4) The number of traffics thus generated was considered as the number of objectives (or problem dimension) and chosen between 5 and 20.
5) The capacities of the networks were all set to a constant value 100. The reason for not taking random values here is simply that otherwise, the test will rather be biased by the minimum value of random capacities. Such a minimum capacity will already act as a bottleneck to a single traffic alone, and thus reduce the influence of link sharing on the fairness situation.

TABLE I
DIMENSION-DEPENDENT PARAMETER.

dimension	5	10	15	20
population/swarm size	20	50	100	200

Using the Floyd algorithm on random link weights to generate random traffic has the advantage of 1) generating also paths of middle size, as the path lengths are not required to be short; 2) enforcing congestion, as links with smaller random weights will be more likely to be used by more than one path for minimizing the sum of weights over the whole path.

2) Algorithm Design: Four algorithms (SPEA, NSGA, PAES, MOPSO) were designed following the principles given in section II-B. All algorithms basically followed the original EMO implementation, with the named adjustments in order to handle fair dominance relation, and the only differences in the design were as follows:

1) The archive was not used in case of SPEA and NSGA. Reason is that if using Pareto-dominance relation to fill the archive, in the case of many objectives the archive starts to virtually accept each individual. Corresponding countermeasures for archive oversize were not in the scope of this study. If the fair dominance relation was used to fill the archive, no reasonable results could be achieved during initial tests.
2) Each individual was a vector of traffic amounts for the particular sender-receiver pairs. Usually, an individual will not represent a feasible state, and it was repaired by the linear congestion control. For repairing, for each sender-receiver pair, the minimum possible traffic after all nodes affected by the traffic selection performed linear congestion control was replacing the current traffic value. The individual stayed in the population in the repaired form. This repair has to be seen as counterpart to the fitness computation in "common" optimization.
3) Individual components were initialized to random values between 0 and 100, and further modifications were clamped between 0 and 200.
4) Standard uniform mutation was used, where a strength parameter described the standard deviation of a modification with respect to the size of a component. Uniform cross-over was used as cross-over operator.
5) In case of MOPSO and PAES, the evaluations related to crowding in feasible space were simplified to uniform random selections (crowding in higher-dimensional spaces is a rather unpredictable factor).

Other relevant parameter settings are given in tables I and II.

3) The Evaluation: The tests were performed for various numbers of objectives. Given this value n, such a number of paths was randomly selected from a fully connected graph of 6 nodes, and the fairness state was computed using the FSM

TABLE II
DIMENSION-INDEPENDENT ALGORITHM PARAMETER.

All	
mutation probability	0.3
max init value	100
value range	0..200
max number of repairs	100000
SPEA	
mutation strength	0.5
crossover scheme	tournament
NSGA	
mutation strength	0.5
crossover scheme	tournament
PAES	
mutation strength	0.001
MOPSO	
inertia weight	0.99
weight p_{best}	0.8
weight p_{leader}	0.3
mutation ratio	0.1
topology	full
mutation strength	0.5

TABLE III
AVERAGE DISTANCES FROM FAIRNESS POINT FOR INCREASING DIMENSION.

	5	10	15	20
SPEA	1.05	15.0	36.0	60.1
NSGA	35.0	62.3	63.1	78.3
PAES	0.02	31.8	63.2	87.0
MOPSO	0.52	10.4	42.6	65.1

algorithm. Then, the four FMAs were tested for each such-generated test problem. In order to have a direct comparison of the performance and despite of the different designs, the maximum number of repairs of traffic amounts (what reflects fitness evaluation in common optimization) was set to 100000. After reaching this limit, the algorithms could finish the current generation, and the average Euclidian distance of the population/swarm to the fairness state was taken as numerical result of a single run. This test was repeated for 30 random selections of traffic, in order to achieve statistical relevance.

B. Results and Discussion

The results are summarized in table III. There, the average distances over 30 runs are shown. Obviously, the four algorithms perform differently. In general, all algorithms are capable of approaching the fairness state, but NSGA design principle shows worst performance, while SPEA and MOPSO show best performances. The performance of all algorithms falls strongly around 10 objectives. For a better understanding of the precision of approximation, take the case 10 objectives and assume a constant offset of individual components from the corresponding components of the fairness state. This is then about a third (approx. $1/\sqrt{10}$) of the Euclidian distance, and a value of 10, for example, indicates an average difference to individual components of about 3 - sufficiently close in a practical sense (note that the values of the fairness state components here range between 10 and 100). For further illustration, in fig. 2 also the distribution of distances for 30 runs in case of 10 objectives is shown.

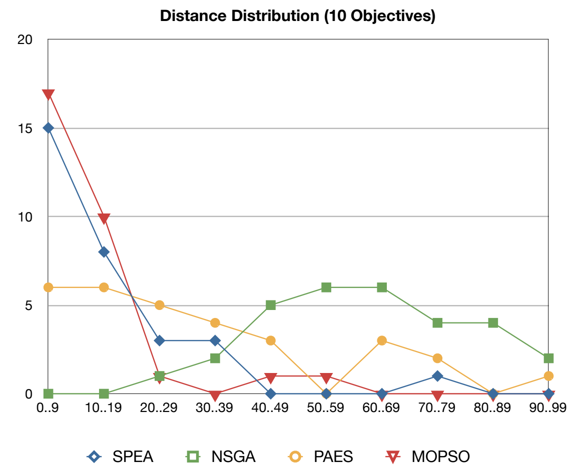

Fig. 2. Distribution of distances from fairness point for the four considered algorithms and 10 objectives.

Here, it can be seen that the distribution of NSGA results more resembles a normal distribution, while SPEA and MOPSO have the majority of runs very close to the correct fairness point. PAES, which gives excellent results in case of 5 objectives, falls rapidly in performance for larger number of objectives, despite of still having a roughly monotonically decreasing distribution of average distances. We will discuss the algorithms in more detail in the following.

1) SPEA: While for lower number of objectives the result is not the best, the performance stays to be among the best for increasing number of objectives. Thus, it can be said that the design principle of SPEA has highest sustainability. Figure 3 shows an example for the evolution of average distances. There is a visible "plateau" structure: the algorithm maintains a nearly constant performance over a number of generations, before it drops to a lower level. We explain this effect with a model of "volume safeguarding" of the feasible space. Due to the used selection criteria, the individuals are not required to belong to the maximum set of the maxmin fair dominance relation of the current population. An algorithm based on Pareto dominance relation can take advantage of its transitivity, and shift its Pareto set towards the real Pareto front of the search problem. This is by way of guarding all inner parts of the current Pareto front by a manifold of one dimension less. In case of incomplete transitivity, it is not sufficient to maintain a front, but also to safeguard the whole inner part, i.e. a manifold of same dimension. The results indicate that this is actually achieved by the SPEA algorithm design principle, also showing the appearance of temporarily stable individual placements.

2) MOPSO: Also MOPSO is steadily among the best algorithms. Figure 4 details an example run of this algorithm. It does not show any plateaus like SPEA, but a rather continuous decay towards smaller distances to the fairness state. The performance drops for larger number of objectives. The MOPSO way of progressing can be rather understood as

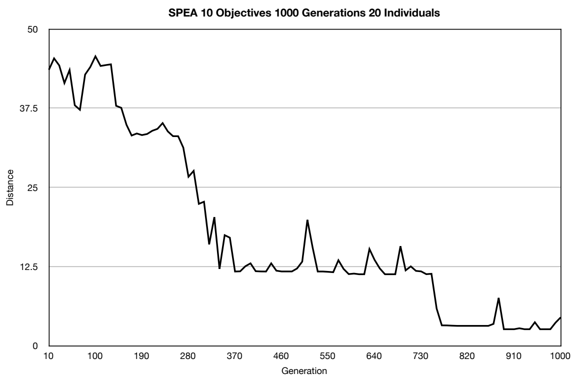

Fig. 3. Values for distances from fairness point for an example run of SPEA (10 objectives, 20 individuals, 1000 generations).

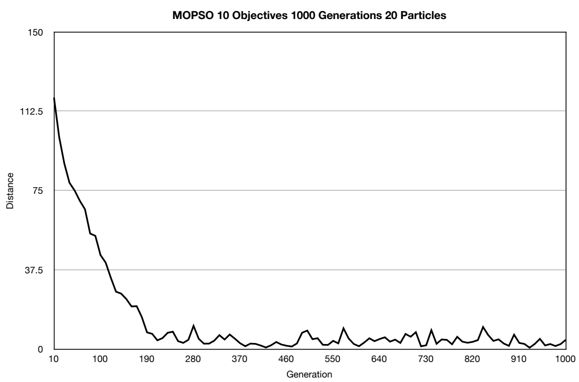

Fig. 4. Values for distances from fairness point for an example run of MOPSO (10 objectives, 20 particles, 1000 generations).

following a stochastic gradient in rather small steps (probably combined with safeguarding by the local best positions of the particles), and it might be related to a stronger effect of mutation than velocity update (subject to further investigations) on the performance.

3) PAES: PAES shows a nearly perfect result for 5 objectives, but then looses heavily. In comparison, it should also be mentioned that the PAES processing has lowest complexity, and it is the fastest algorithm among the four. It seems that the safeguarding of the randomly produced candidate individuals by the archive works well for smaller number of objectives, but then the archive fails to cover the feasible space accordingly (the point were SPEA seems to succeed much better). Further improvements of PAES for such kind of problems can focus on this aspect.

4) NSGA: Similar to the case of multi-objective optimization, NSGA does not handle the case of many objectives very well. An investigation of algorithm internals (not detailed here) gives that most of the time the population will only contain rank 0 individuals. This is a little bit surprising, as the probability of maxmin fair relation random appearance is not falling exponentially [7]. Indeed, the genetic operators produce children, which are together with the parent generation ranked by values up to 20. Nevertheless, after selection the majority of the population receives very low ranks. Then, the algorithm is forced to rest on the secondary ranking criterion (crowding distance) alone, which is not a good guidance for the evolution.

IV. Conclusion

The results presented in the former section demonstrate that algorithms designed from the same principles as multi-objective optimization algorithms are capable of approximating maximum sets of other relations, like the fair dominance relations. The results also show that there are strong differences between the approaches. The most successful scheme is the comparison of individual points based on the "Pareto-strength" (we use quotation here, as the used relation is not Pareto-dominance but fair dominance), closely followed by the swarm principle. Such results can be achieved even without having the property of transitivity of the used relations (but including having no conflicts but only incompleteness in transitive relations, as otherwise the maximum set might be empty at all). A possible understanding for the better performance can be related to the "safeguarding" of the search space by the population, covering the whole search space explored so far and not only its border as in case of the Pareto-dominance. As an effect related to the measure (hyper-volume) of parts of the search space, thus the performance also rapidly decays with increasing number of objectives. For a number of objectives larger than 20, the exponential increase in search effort (e.g. measured by needed population size and number of generations) might render the approach impracticable. It could be also seen that the concept of ranking, as used e.g. by the NSGA-II algorithm, is not working very well in this context, as most of the individuals happen to stay on the lowest rank. Nevertheless, further studies should clarify why e.g. the concept of PAES fails so strongly for more than 10 objectives, while having nearly perfect results for a smaller dimension. The goal then is to find new design principles for the case of fairness, by fusing the strong and weak points of EMO design principles as indicated by the first results presented here.

Acknowledgment

This work was partly 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 and G. B. Lamont, Eds., *Applications of Multi-Objective Evolutionary Algorithms*. Singapore: World Scientific, 2004, ISBN 981-256-106-4.

[2] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, N: Prentice Hall, 1992.

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

[4] J. Jaffe, "Bottleneck flow control," *IEEE Trans. Commun.*, vol. COM-29, July 1981.

[5] S. Haldar and D. K. Subramanian, "Fairness in processor scheduling in time sharing systems," *SIGOPS Oper. Syst. Rev.*, vol. 25, no. 1, pp. 4–18, 1991.

[6] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared systems," DEC, Tech. Rep. Research Report TR-301, 1984.

[7] M. Köppen, M. Tsuru, and Y. Oie, “Evolutionary approach to maxmin-fair network-resource allocation,” in *Applications and the Internet, IEEE/IPSJ International Symposium on*. Turku, Finland: IEEE Computer Society, 2008, pp. 253–256.

[8] F. Kelly, “Charging and rate control for elastic traffic,” *Eur. Trans. Telecomm.*, vol. 8, pp. 33–37, Jan./Feb. 1997.

[9] X. Wang, K. Kar, and J.-S. Pang, “Lexicographic max-min fairness in a wireless ad-hoc network with random access,” in *Proceedings of IEEE Conference on Decision and Control (CDC)*, San Diego, CA, December 2006.

[10] M. Köppen, K. Yoshida, M. Tsuru, and Y. Oie, “Fairness-based global optimization of user-centric networks,” in *INC, IMS and IDC, 2009. NCM ’09. Fifth International Joint Conference on*, Seoul, Korea, Aug. 2009, pp. 441 –443.

[11] J. D. Schaffer, “Multiple objective optimization with vector evaluated genetic algorithms,” in *Proceedings of the 1st International Conference on Genetic Algorithms*. Hillsdale, NJ, USA: L. Erlbaum Associates Inc., 1985, pp. 93–100.

[12] 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.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimisation: NSGA-II,” in *PPSN VI: Proceedings of the 6th International Conference on Parallel Problem Solving from Nature*. London, UK: Springer-Verlag, 2000, pp. 849–858.

[14] J. Knowles and D. Corne, “The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation,” in *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, vol. 1, 1999, pp. 98–105.

[15] M. Reyes-Sierra and C. A. C. Coello, “Multi-objective particle swarm optimizers: A survey of the state-of-the-art,” *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 287–308, 2006.

[16] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in *IEEE International Conference on Neural Networks, Proc.*, vol. IV, 1995, pp. 1942–1948.