Comparative Study on Meta-Heuristics for Achieving Parabolic Fairness in Wireless Channel Allocation

Mario Köppen, Kaori Yoshida, Rodrigo Verschae, Masato Tsuru, and Yuji Oie
Kyushu Institute of Technology
680-4 Kawazu, Iizuka, Fukuoka 820-8502, Japan
Email: mkoeppen@ieee.org, kaori@ai.kyutech.ac.jp, rodrigo@verschae.org, tsuru@ndrc.kyutech.ac.jp, oie@cse.kyutech.ac.jp

***Abstract*—We present the results of a comparative study on the design of meta-heuristic algorithms for achieving parabolic fairness in wireless channel allocation. Wireless channel allocation (WCA) is a basic problem of fair distribution of indivisible goods, in this case the allocation of channels to users in a wireless schedule. Parabolic fairness represents a state that coincides with maxmin fairness in fair end-to-end user traffic rate allocation. This state can also be represented by maximization of a special case of the ordered weighted averaging operator. The related task then is to find an algorithm to approximate that maximum value for the WCA as close as possible. Here, several heuristic approaches are taken into account: a simple annealing heuristic, its integration into an Iterated Local Search (ILS) and also the integration of ILS as local search of a memetic algorithm. The comparison gives that best results can be achieved by the ILS, with up to 2-3 times improved performance compared to other algorithms and for practically relevant problem dimensions.**

***Keywords*-maxmin fairness; parabolic fairness; ordered weighted averaging operator; wireless channel allocation; meta-heuristics; iterated local search; memetic algorithm**

I. Introduction

Recently, for the design, control and utilization of data and communication networks and distribution of network resources, the awareness for aspects of economics is rapidly increasing. One reason for this can be seen in growing demands in response to expanded opportunities of network services. The natural consequence here is the need to share limited resources in an efficient manner. In this context, the common paradigm of global optimality often becomes infeasible, esp. in circumstances where subjective valuations interfere with abstract objective measures of network performance. This can refer to situations where users are virtually exempted from resource access for the "sake of global optimality." Economics is offering a rich palette of concepts (like envy-freeness, Pareto stability, utility, prospect, fairness, equity) to avoid or circumvent such situations, but the offer of practical tools to achieve such goals is rather limited. For example, with regard to equity, we could consider fair division "cake cutting" concepts to allocate traffic rates in a wired network and end-to-end user traffic under link-capacity restraints, but this approach needs to specify utilities for each user and each traffic rate. Usually, there is neither the opportunity, nor the time to question all users about their opinion about a traffic rate, and also, it might be hard for a human to assess fine-grained numerical values. But without these values, equity cannot be achieved. Similar problems arise for other related concepts. A notable exception here is the allocation of traffic rates by the Bottleneck Flow Control algorithm (BFC) [1]. The final state achieved by this algorithm can be characterized in several ways. With regard to uniqueness, the algorithm gave raise to the definition of *maxmin fairness*, and also it is known to maximize the minimal rate allocated to any user. Recently, another unique characterization of the final state of the BFC algorithm, called parabolic fairness, has been introduced [2]. There, it was shown that a functional value, calculated from user traffic rates alone (i.e. without reference to the link capacities) exists, which is maximized by the BFC allocation, and that there is no other rate allocation with a larger or equal value. The function is a special ordered weighted averaging operator with anti-ordered and exponentially increasing weights - expOOWA for short. For above mentioned traffic rate allocation problem, the state with maximum expOOWA value is exactly the maxmin fair state. This allows for a convenient extension of the concept of "BFC-fairness" to other domains, by direct function maximization and without the need to provide a logical counterpart for other logical concepts of the BFC algorithm like link capacities and the procedure of "flooding."

Traffic rate allocation is sharing of an (infinitely) dividable resource, and also, traffic rates can be directly manipulated. Many other problems in networking do not show this characteristic, and we rather find individable resources to share, and performance measures that cannot be directly manipulated. These problems are usually of discrete and combinatorial nature. The allocation of carrier channels to mobile agents for wireless network access can be seen as a basic problem here, since it represents the essential combinatorial aspect of a number of recent challenging problems for the operation of wireless networks. Currently, IEEE 802.16j task group is working on relay-based multi-hop wireless access standards. Also here, the major aspect that influences efficiency is a combinatorial task of relay station allocation to mobile agents (cooperative relaying) together with modality choices for the relays (whether amplify-and-forward or decode-and-forward mode), and sub-carrier allocation to mobile agents in OFDMA, i.e. all facets of distribution of indivisible goods. But then, they also mix with "traditional" continuous optimization problems like power allocation to stations with regard to achievable SNR and relay placement, often under QoS or territory constraints, and we can define an endless number of constrained optimization problems (see [3] for a nice introduction).

The Wireless Channel Allocation problem (WCA), where the focus is on the task of assigning channels over a number of time slots to mobile agents ("users") and by fusing the various infrastructure and physical aspects of channel transmission for each user, timeslot and channel into a single numerical value, a so-called *channel coefficient*, has already gained some attention, and several approaches have been proposed. Examples include the use of bidding or auction systems [4], payment systems, we find related game-theoretic approaches, or approximations to maxmin fairness [5]. However, the direct application of such concepts to network configuration and operation remains a challenging task. Therefore, here we focus on the more direct approach to define a computable criterion for efficient manipulation represented by parabolic fairness and the corresponding expOOWA maximization. In [6] it was already investigated how parabolic fairness fits into the frame spanned by other well-studied fairness relations like maxmin fairness, leximin fairness, or proportional fairness. However, exhaustive search of all possible allocations to find the maximum state(s) for such relations becomes impossible even for smaller problem scales due to combinatorial explosion, and also here, exact and tractable algorithms are not yet known. This is a good chance to study the use of heuristic and meta-heuristic approaches in more detail. In [6], a simple annealing heuristic based on properties of the expOOWA was already presented, with some chance to achieve good results, but also some risk to "drift away" from the true maximum state. Here, we want to consider the option to wrap this heuristic into a conservative framework algorithm, and study several design alternatives, in particular Iterated Local Search and Memetic Algorithms.

This paper is organized as follows: next section will recall the needed concepts and definitions. In Section III the various design alternatives of a meta-heuristic approach to parabolic fairness are studied. Corresponding results are summarized in Section IV.

II. Terms and Concepts

A. Wireless Channel Allocation

In Wireless Channel Allocation, a blank matrix B of channel-timeslot pairs with a total of M cells b_i, a set $U = (u_i)$ of N users and an $M \times N$ matrix C of *channel coefficients* of real values from $[0, 1]$ are given. The task is to enter at most one user into each blank cell in B, i.e. to provide an *allocation* $a : B \to U$ of cells to users with $|\{u \in U \,|\, a(b) = u\}| \leq 1$ for all cells $b \in B$. Each entry c_{ij} of the matrix C represents the *utility* for user u_i in case of assignment of cell b_j, as a model abstraction of all the physical and logistic circumstances of the wireless access. For a given allocation a, the *performance* for each user is given by

$$p(u_i) = \sum_{j, a(b_j) = u_i} c_{ij} \tag{1}$$

i.e. the sum of channel coefficents for all channels allocated to the same user. Channel allocation has to be performed such that, in some sense, all users are "satisfied" with their individual performances as good as possible. The actual problem is to specify the meaning of "satisfied" in an efficient way. For example, considering maximization of the sum of all performances is not a good way to satisfy all users: the optimization problem could be easily solved by selecting for each cell one of the users with a maximal channel coefficient. But this way it can happen that then some users will never get any channel allocated, and these users are exempted from wireless access. Therefore, the economics of WCA becomes relevant, especially aspects of fairness.

B. Parabolic Fairness

Within the feasible domain of all end-to-end user traffic rate allocations in a network with link-capacity constraints (i.e. linear inequality constraints), the maxmin fair state is characterized such that for each user getting better off there is at least one other user, already equal or worse off, that must become even more worse off. This state of traffic rates can be achieved by the BFC algorithm. In [2] it was shown that this state can be exactly characterized by the maximization of a special ordered weighted averaging operator. In general, if there is a set of n real numbers x_i with $i = 1, \ldots, n$ and a set of n real-valued weights w_i, and if (i) as a sub-script indicates the i-th smallest element of an ordered set, then the ordered-ordered weighted averaging operator is defined as

$$OOWA_w(x) = \sum_i w_{(i)} x_{(n-i+1)} \tag{2}$$

which means that we compute a weighted average of the x-values where the largest value is weighted with the smallest weight, the second-largest value with the second-smallest weight etc. For representing the maxmin fair state for any set of link-capacity constraints, it is necessary and sufficient that the weights also increase exponentially, i.e. the sum of the i-th smallest weights is smaller than the $(i+1)$-th weight. In this case, we use the notation $expOOWA_w(x)$. A convenient choice for the weights is $w_i = 2^{(i-1)}$.

Parabolic fairness then is the transfer of this concept of maxmin fairness to other domains in the sense of analogy, as the literature style of a parable does, by imposing expOOWA maximization as the means of efficient state allocation. Then, if we consider the user's performance vector, we can introduce a concept of maxmin fairness in the WCA problem as well.

C. Annealing Heuristic

In [6] also an annealing heuristic for maximizing the expOOWA value of a WCA problem was studied, which was based on mathematical properties of the expOOWA operator. For a given allocation, two modifying operators where tested whether their application fulfills a given condition. If the condition is fulfilled, the modification is accepted. Even in case the criterion is not fulfilled, with a small *annealing* probability it is accepted as well. Otherwise it is rejected. The two operators and their conditions are:

1) the REPLACE operator, where for a given allocation of users to cells, a random cell will be selected and its assigned user replaced with a randomly selected different user. If the average of the performances of both users will not decrease, and the absolute difference will decrease, the REPLACE is accepted.
2) the SWAP operator, where we select two cells with different user allocation at random, and swap their user allocation. If this increases the performances for both users, the SWAP will be accepted.

D. Iterated Local Search

Iterated Local Search (ILS) is a recent meta-heuristics that utilizes a local search to handle the optimization problem at hand. The ILS concept is a generic one, as it needs only to specify the so-called "local heuristic" and a modality of its repeated application. Assume we have such a local heuristic that gets stuck into a local optimum after a number of steps. Often it is observed that instead of increasing the number of iteration steps it is more efficient to restart the algorithm from a different initial position in the search space, but keeping the same (smaller) number of iterations. This observation became a rule for the ILS: after applying the local heuristic for a number of iterations, a new starting point is derived, and the local heuristic is started again.

One feature in this regard that distinguishes ILS from e.g. stochastic hillclimbing is the iterated application of the local search, i.e. it first modifies the currently known local extremum, then modifies the modified one etc. Thus, the algorithm has a better chance to progress much further in the searchspace, depending on the probability that a better solution is nearby known good solutions.

Thus, the efficiency of this algorithm can vary strongly, depending on configuration settings. However, the high genericity and flexibility, easy implementation and set up, and also successful application especially for combinatorial optimization problems makes ILS a good and straightforward candidate algorithm to "wrap" the formerly studied annealing heuristic.

E. Memetic Algorithms

Memetic algorithms refer to a family of meta-heuristic search algorithms, where an evolutionary or other population based approach (like swarm intelligence) is enhanced by integrating a separate individual (particle) learning or local improvement procedure (local search). The most simple case here is the addition of local search for better solutions by each parent in a genetic algorithm. ILS an be used as such a local search procedure.

III. Algorithm Design Experiments

Often, results of the application of heuristic or meta-heuristic algorithms are not predictable, or only rough bounds can be specified, and are known to be sensitive to parameter settings and to algorithm configuration. So the question about a good approach can usually be answered only experimentally. Here, we will follow a procedure to design a good meta-heuristic for maximizing the expOOWA operator for a random set up of a WCA problem, in order to find a fairmost allocation of channels to users. The annealing heuristic will be integrated into an ILS, and the best performing ILS configuration then will be integrated into a memetic algorithm.

A. Configuration of Iterated Local Search

We recall that we are considering two local modifications, REPLACE and SWAP. If a random modification fulfills an indicator criterion, the modified solution (here a user's allocation to channels) is accepted. If not, it is usually refused, but with some probability (the annealing probability) it might be accepted as well.

We will formally indicate a configuration of ILS as $ILS(n + m, N, p_1, p_2)$ with the meaning: the Local Search is composed of n-times application of REPLACE with annealing probability p_1 and m-times application of SWAP with annealing probability p_2. After $n + m$ applications, the solution with the highest expOOWA value replaces the former maximum solution. Then, this Local Search is iterated N times.

Different settings for these values were tested for the WCA problem with 10 users and 12 cells. Each time, the ILS algorithm was tested on 30 random instances of the WCA problem, and the values of the ILS were set such that the total number of samples remained constant (i.e. $N(n + m) = 10,000$) to allow for a fair comparison. We took the set up $ILS(5000 + 5000, 1, 1, 1)$ as "master configuration" and divided the performance of the other ILS variants by the performance of this ILS (for each random instance of the WCA separately). This ILS is in fact not an ILS but the plain application of SWAP and

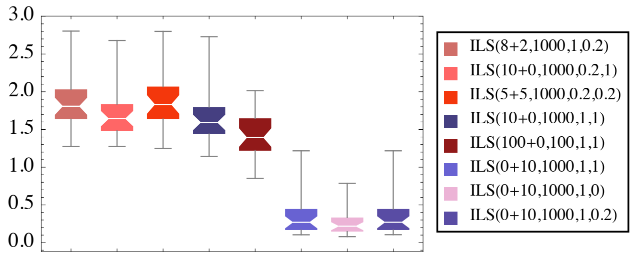


Figure 1. Results for various configurations of Incremental Local Search (ILS). See text for the meaning of algorithm formulas.

REPLACE without any repetition of the Local Search, and without annealing. The results can be seen in the boxplots in Fig. 1. There, the y-axis value 1 corresponds to the $ILS(5000+5000,1,1,1)$ performance.

The first observation are strong differences in performance for various settings. In more detail, we can see:

- ILS using SWAP only perform rather weak (last three box plots in the figure), while ILS with REPLACE only (boxplots 2, 4 and 5) perform better than the master ILS.
- ILS with SWAP only performs (slightly) better when using annealing (here the value 0.2 in boxplot 8).
- ILS with REPLACE only perform better for smaller number of Local Search iterations (boxplot 5 shows the drop in performance when increasing this number from 10 steps to 100).
- The annealing has no notable influence on the performance for ILS using REPLACE alone (compare boxplot 2 and 4).
- While using SWAP heuristic alone is deteoriating performance, mixing it with REPLACE gives an increase in performance (boxplots 1 and 3).

From this comparison, we can see that a set up of ILS as $ILS(5+5,1000,0.2,0.2)$ is giving best results for the application of ILS.

B. Memetic Algorithm

As mentioned before, we can try to even further increase performance by integrating the ILS into a Genetic Algorithm (GA) as local search procedure. However, also here, we can consider different ways of doing such an integration. Formally, a GA can be seen as population based search, starting with a random initial configuration of N_p parents. Then, from parents, the same number of children is generated by application of genetic operators, usually (fitness-related) cross-over and (contingent, i.e. fitness-independent) mutation, and from the union of parents and children, after another evaluation by the specified fitness function, the N_p best individuals are selected to become the next parent generation. So, the "formula" for a standard GA can be written in operator notion as $parents + children(parents)$ to specify the construction of the pool from which the individuals of the new generation are selected.

If we integrate another operator, called $memetic()$ we have to specify which pool undergoes the local search, and how it is combined with the other operator results into the selection pool. Then, we can consider the following 5 more relevant modifications:

Mode 1: $parents + children(memes(parents))$
Mode 2: $parents + memes(parents)$
Mode 3: $parents + memes(par.) + children(par.)$
Mode 4: $parents + memes(children)$
Mode 5: $memes(parents) + children(parents)$

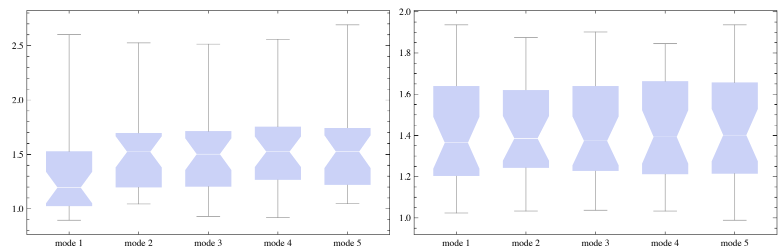
For example, in Mode 1 we apply ILS to the parents, and then the standard GA operators in order to generate children, which are combined with the parents into the selection pool.

Also here, we apply the different versions of the memetic algorithm to random instances of the WCA problem for 10 users, 12 cells, in each case repeating 30 times. The ILS was the one showing best performance in the experiments that were performed before, i.e. $ILS(5+5,1000,0.2,0.2)$. There were no changes applied to the GA parameters itself: $N_p = 10$, using tournament selection and polynomial mutation with exponent 2.5. The number of iterations was set in a manner to ensure that each modification in total evaluates 11010 positions in the search space (note that this odd number comes from the fact that each modality evaluates a different number of individuals, and also to make sure to have integer values for the number of evaluations).

Corresponding boxplots are shown in Fig. 2(b), after dividing the maximum value of the exponential OOWA found in corr. memetic mode with the maximum value yielded from applying the standard GA. Then, figure 2(a) shows the same for 1101 evaluations of search space positions (i.e. about 100 generations). The evaluation of this result is rather simple: all modifications perform much better than the standard GA (about 1.5 times), but the differences among them are rather minor. Only mode 1 seems to evolve more slowly towards the maximum value than all other. A slight advantage can be seen for mode 4, where the memetic search is applied to the children only. In lack of any stronger distinguishing criterion, we will go ahead with this modality of the memetic algorithm.

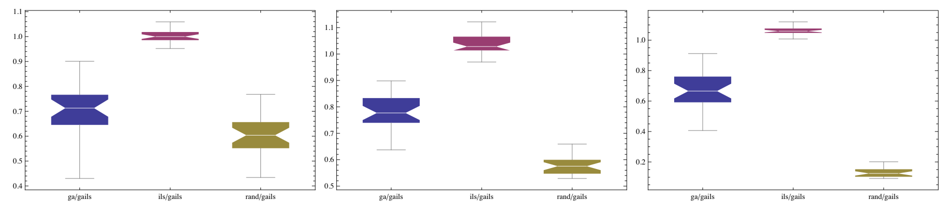
C. Algorithm Comparison

As a result of the foregoing subsections, we consider the set up $ILS(5+5,1000,0.2,0.2)$ as local search in the memetic algorithm $parents+memes(children)$ (*gails*) and want to compare the performance of this algorithm with the "standalone" ILS (*ils*), the GA without ILS (*ga*) and also with random search (*rand*). All four algorithms were tested on random instances of the WCA problem for 10 users and 12 cells, 10 users and 30 cells, and 30 users, 50 cells. The base performance (value 1.0) was computed by using the



(a) 100 generations (b) 1000 generations

Figure 2. Boxplots of performances achieved for the various modes of integrating ILS into GA.



(a) 10 users 12 cells (b) 10 users 30 cells (c) 30 users 50 cells

Figure 3. Comparison of performances for increasing problem dimension, in each sub-figure from left to right: GA, ILS, and random search performances, each divided by the performance of GA with imputed ILS.

memetic algorithm *gails*. Again, 30 random instances of the WCA problems were tested in each case.

The results can be seen in Fig. 3. Some observations:

- The memetic algorithm *gails* does not perform better than *ils*. An even slightly better *ils* performance can be observed for increased problem scale.
- The standard GA is always outperformed by *gails*, and providing about 0.7 times smaller maximum values of the expOOWA.
- For smaller problem dimension, *ga* is even closer to random search, but "matures" from random search for larger problem dimensions. The related observation: random search becomes highly inefficient for larger problems.

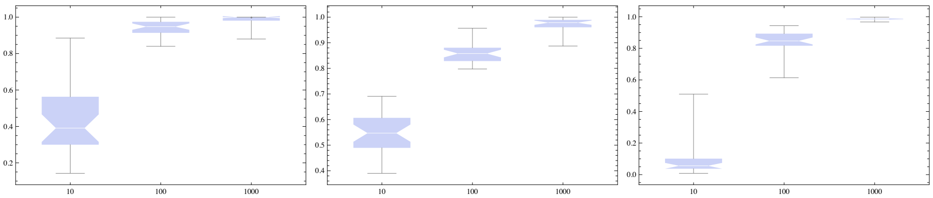
In summary, whether used alone or in combination with a GA, the most efficient approach as a meta-heuristic appears to be the ILS. An explanation of the weaker performance (or better lack of any notable improvement) of *gails* might be seen in problems with the encoding of an allocation into an individual and the corresponding "large jumps" in search space as an effect of the application of genetic operators. While testing the ILS, it was obvious that the SWAP heuristic, affecting TWO positions, already gave a strong drop in performance, compared to the REPLACE heuristic, affecting only ONE position. A related simple experiment not reported here in detail confirmed the tendency of performance decrease when affecting more and more positions in an allocation. We think that the GA operators are doing exactly this, affecting several positions, and this is resulting into a rather weak performance that is not fully captured by the more sophisticated interplay of the genetic operators.

D. Other Evaluations

Finally, having found that an ILS using 5 REPLACE and 5 SWAP, both with annealing probability 0.2 is giving the most promising performance values, we want to see a few other aspects of using this algorithm. One aspect is related to potential real-time use of the algorithm. We want to consider the trade-off between number of iterations (N) and performance loss by using a smaller number of evaluations. This is shown in Fig. 4 for the same problem dimensions as in the foregoing section, and repeated 30 times, but using only 10, 100, and 1000 iterations of Local Search.

The base value 1 here is the performance after 10,000 iterations. It can be clearly seen that for all problems, nearly the same performance is already achieved after 1000 iterations, so from practical point of view the remaining 9000 iterations do not really provide a gain in performance. The question whether 100 iterations might also suffice is related to an estimated 10% drop in performance (but note that in none of these cases we know the exact maximum value, as the huge size of the search spaces forbids any attempt of an exhaustive evaluation). In any case, 10 iterations will never provide a suitable performance, except a few "lucky" cases for the smaller problem size.

Finally, we want to investigate the question if all these evaluations are related to the use of parabolic fairness at all. Figure 5 shows a similar comparison of performances of *ga*, *gails*, *ils* and *rand*, i.e. for 10 users, 12 cells, but



(a) 10 users 12 cells (b) 10 users 30 cells (c) 30 users 50 cells

Figure 4. Illustration of progress of ILS for 10, 100 and 1000 local evaluations vs. 10000 evaluations for increasing problem dimension.

using the minimum performance of a user in an allocation as the fitness function that we want to optimize. Also here, it seems that *ils* alone performs best, and also nearly same as *gails*, but there are two notable differences: (1) the extreme value of minimal *ils* performance can be 0! This refers to allocations that do not contain some of the users, and cannot be easily repaired by the local heuristics, if more than one or two users are not appearing in the allocations. On the other hand, from only comparing the minimum performances, there is no guidance towards allocations including more users. In comparison, changing allocations always affects the value of the expOOWA numerically, and when using this measure, the algorithm can escape this "trap." (2) The performance of standard GA is very likely that of a random search. Also here, the explanation is the same: the leximin measure does not acknowledge changes apart from the user with minimum allocation, and a heuristic search is prevented from progress. This small experiment also confirms the better suitability of parabolic fairness with regard to heuristic search guidance.

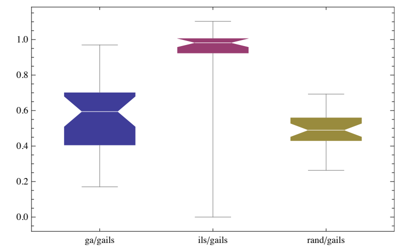


Figure 5. Comparable result for using leximin relation instead of the expOOWA (same order as in Fig. 3).

IV. Conclusions

In this paper, we have investigated various designs for meta-heuristic algorithms to approximate the parabolic fairness state in Wireless Channel Allocation. The optimization goal here is to maximize a special ordered weighted averaging operator, called expOOWA. Focus was given to the stepwise integration of simpler heuristics into meta-heuristics. As a result, in a hierarchical sense, a plain annealing heuristics alone appears to be incapable of handling this optimization task, while its integration as local heuristic into an Iterated Local Search algorithm then performs better than the annealing heuristic, as well as the integration of the ILS itself into a memetic algorithm. We can confirm the good performance of an ILS using a smaller number of local steps (here 10), along with a rapid approximation already within 100-1000 iterations of the local search for technically relevant problem scales. Note that this is a general results since we did not make any specific model assumptions for wireless transmission. The necessary continuation of this work is the study of further opportunities of parabolic fairness and related expOOWA maximization under limited knowledge of the current network situation.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 24650030. One of the authors was supported by the BecasChile Postdoctoral Scholarship Program.

References

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

[2] M. Köppen, J. Okamoto, and A. Honda, "Fuzzy fusion fairness relations for the evaluation of user preference," in *2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Proceedings*, Taipei, Taiwan, June 2011, pp. 2566 – 2574.

[3] D. Zhang, Y. Wang, and J. Lu, "On QoS–Guaranteed Downlink Cooperative OFDMA Systems with Amplify–and–Forward Relays: Optimal schedule and resource allocation," in *2009 IEEE Wireless Communications and Networking Conference*, April 2009, pp. 1–5.

[4] J. Sun, E. Modiano, and L. Zheng, "Wireless channel allocation using an auction algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 1085–1096, 2006.

[5] L. Tassiulas and S. Sarkar, "Maxmin fair scheduling in wireless networks," in *Proc. IEEE INFOCOM, New York*, 2002, pp. 763–772.

[6] M. Köppen, K. Yoshida, M. Tsuru, and Y. Oie, "Annealing heuristic for fair wireless channel allocation by exponential ordered-ordered weighted averaging operator maximization," in *2011 IEEE/IPSJ International Symposium on Applications and the Internet (SAINT 2011), Proceedings*, Munich, Germany, July 2011, pp. 538–543.