Annealing Heuristic for Fair Wireless Channel Allocation by Exponential Ordered-Ordered Weighted Averaging Operator Maximization

Mario Köppen, Kaori Yoshida, 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, tsuru@ndrc.kyutech.ac.jp, oie@cse.kyutech.ac.jp

***Abstract*—Wireless channel allocation (WCA) is a relevant resource allocation problem that cannot be handled by a simple maximization approach. In such a case, users can often be excluded from receiving channels at all. Therefore, here we want to focus on fairness concepts for specifying wireless channel allocations. We extend the concept of fairness in traffic congestion avoidance to the wireless channel allocation problem. The main characteristics of the state achieved by the fair Bottleneck Flow Control algorithm were directly applied to the WCA problem, including lexmin fairness, maxmin fairness, proportional fairness, and, as a new result, exponential Ordered-Ordered Weighted Averaging (OOWA) operator maximization. The utility of the numerical exponential OOWA maximization for reflecting fairness in the WCA is demonstrated. Based on this and two basic properties of the general OOWA, a *replace* and *swap* operation based annealing heuristic is proposed. Its study on small problem instances allows justifying their applicability to real-world instances of the WCA problem.**

***Keywords*-fairness; maxmin fairness; proportional fairness; ordered weighted averaging operator; wireless channel allocation; bottleneck flow control**

I. Introduction

The wireless channel allocation (WCA) problem refers to the allocation of channels for a sequence of consecuting time slots to a numbers of users by a base station in a wireless communication network. For each channel, time slot, and user, there is a so-called channel coefficient, representing the physical characteristics, by which a user can send data through the channel at the specific time slot. Usually, these values are predictions and are drawn from distributions according to a physical model of the wireless infrastructure. Seen as a combinatorial problem, and abstracting from the specific set up, the conditions of the WCA can be given as follows. A set of m channel-timeslot pairs, which we will call here *cells*, has to be assigned to a set of n users u_i. Each user is specified by a vector c_i of m channel coefficients ($m \geq n$). An allocation a is an index vector of size m, where the index a_i indicates the assignment of the cell i to user u_{a_i}. The performance p of an allocation is the vector of sums of channel coefficients for all users: the component p_i equals $\sum_{\{j|a_j=i\}} c_{ij}$.

The question for this problem is how to decide for an allocation in a specific situation. It can be easily seen that a maximization criterion is not helpful. Actually, it is trivial to maximize the total performance for such a problem, by just selecting for each cell the user with the highest channel coefficient. However, for such a selection it can happen that a number of users will be neglected, once they never have the maximum channel coefficient anywhere. So it also needs some means to ensure that users never become neglected or, in other words, that all users are treated equally. So, we will focus on fairness in Wireless Channel Allocation here.

Fairness in WCA has only rarely been discussed. So far, the focus seems to rather employ game theory, especially bidding concepts [1], or scheduling policies [2], instead of providing exact algorithms. In [3], Bottleneck Flow Control (BFC), an algorithm taken from network traffic congestion avoidance, was proposed as a means for also defining and achieving fairness in the discrete context of the WCA. There, so-called “equality bottlenecks” were identified to represent fair allocations, and a heuristic approach was presented to find these bottlenecks. In this paper, we want to advance on this concept by introducing a new characteristic of the BFC and using it for defining a fairmost allocation of users to cells. The new characteristic is a real-valued function of the performances alone, the so-called exponential Ordered-Ordered Weighted Averaging (OOWA) and it was recently shown that this function is maximized by the BFC algorithm [7]. So, the prospect of using the exponential OOWA also for the WCA will be further explored in this paper, and based on some mathematical properties, an annealing heuristic is derived that can handle the WCA problem under real-time conditions.

Next section will recall the BFC algorithm and its characterization by maximum and best sets of relations, and introduce the exponential OOWA. Section III will provide a study of the same relations for the WCA problem and provide the means for the design of a heuristic approach to fairness in WCA. Section IV will provide the new algorithm and some results.

II. Bottleneck Flow Control Approach to Fairness

While comparing different solutions to a resource sharing task, usually, subjective justification of fairness can be easily done. Doing the same formally and, at the same time, effectively appeared to be not that easy. A notable exception here was already devised about 30 years ago [4]. The studied problem was to allocate traffic volumes to users in network routing, once all paths for the traffic routing through the network were decided. The so-called "bottleneck flow control" was proposed as a means to avoid congestion while being fair to all users as well. The limitation in resources is expressed by assigning maximum capacities to the links in the network. Thus, if the paths do have some overlap in the used links, certain subsets of users will attempt to share links for sending their traffic. If the situation is rather visualized in analogy to a network of water channels, where each user wants to achieve a certain height-level of water along her path, then the algorithm starts with flooding all channels increasingly from level 0. At some point, and for at least one link, a bottleneck will appear, i.e. the rising water level will hit the maximum level for one channel (the maximum capacity in terms of the network problems). Up to this moment, each user has received exactly the same height-level. But obviously, the steady increase in water level cannot be continued anymore, as this will exceed at least one capacity. On the other hand, in most cases, only a subset of users will be affected by this bottleneck. So, the approach is to stop the continuous increase of water levels for the affected users, but continue for all others. The level will continue to raise except for the channels that have reached their maximum level, and eventually reach another bottleneck. Then, the affected users are excluded from further increase as well, and the procedure iterates until all bottlenecks are reached. More details about the algorithm can be found, for example, in [3].

A. *Characterization of the BFC Result*

Given an instance of a network routing problem with maximum capacities, the BFC algorithm will always assign a unique traffic state to all users. Within the feasible space of all possible traffic states, this state T^* can be characterized in several ways. A number of ways have been reported in the past, and all of them lead to a notion of fairness. Here we will focus on four of them.

We also want to recall that for any relation $>_R$, formally seen as a subset of $A \times A$, where A is an arbitrary but fixed set, we can define a *maximum set* and a *best set*: the maximum set contains all elements x from A such that there is no other $y \neq x$ in A with $y >_R x$. The best set, on the other hand, contains all elements x of A such that for any other $y \neq x$ $x >_R y$ holds.

The most simple and obvious relation is the *lexmin* relation. The lexmin relation is defined as a relation between two points from R^n. Given two vectors x and y from R^n, we consider the coordinates of both vectors sorted in increasing order. It is said that $x >_{lm} y$ if and only if the first coordinate of x, in this sorting, which is different from y, is larger than the corresponding coordinate of y in this sorting. From the design of the BFC algorithm (actually already for the first bottleneck), it can be easily seen that state T^* is the best element of the feasible space of all traffic allocations for the lexmin relation.

Second we note the *maxmin fair dominance* relation:

Definition 1. *An element x of the feasible space (a subset of R_n) 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 another component x_j of x which is (already) smaller than or equal to x_i and such that y_j is smaller than x_j.*

It has already been reported (e.g. [5]) that the state T^* is the maximum set as well as best element of this relation. For some weaknesses of the maxmin fair dominance relation (see [6]), several extensions have also been considered in the past, with the *proportional fair dominance* relation being the most popular:

Definition 2. *A point x of feasible space (a subset of R_n^+) is considered proportional fair dominating another feasible point y, if and only if*

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

holds.

Proportional fairness can be seen as an approximation to maxmin fairness but it is not directly related to the BFC algorithm.

While the former definitions show how to use a relation between points in the feasible space, and to characterize the BFC algorithm as the one that finds the best element (or maximum set elements) of these relations, the question comes up if there is also a scalar function of the traffic amounts that is directly maximized by the BFC algorithm. We define a special case of the ordered weighted averaging (OWA) operator. As a reminder, the OWA of a point x of R_n, given a weight vector $w \in R_n$, is defined as $\sum w_i x_{(i)}$. In this expression, $x_{(i)}$ indicates the i-th smallest element of all coordinates of x. We specialize the OWA by also requiring the weights to be sorted in the opposite order:

Definition 3. *Given a point x from R_n and a set of weights $w \in R_n$, the Ordered-Ordered Weighted Averaging (OOWA) of x by w is defined as*

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

Thus, in the OOWA, the largest value is multiplied with the smallest weight, the second-largest value with the second-smallest weight etc. As a special case of the OOWA, we also introduce the exponential OOWA. The additional requirement here is $w_{(k)} > \sum_{i=1}^{k-1} w_{(i)}$, so that the weights itself are exponentially increasing. A possible choice is $w_i = 2^{(i-1)}$ for any $i \in N$.

Based on this, we can state the following:

Theorem 1. *Given a weighted graph G of a network, a routing (i.e. a set of linking paths between nodes), then among all feasible traffic allocations to the users, the BFC algorithm gives the state with the maximum value of the exponential OOWA (for any fixed choice of weights).*

The result has recently been reported in [7]. We also employ two extensions of the result. The requirement of an exponential increase of the weights is a result of link sharing. The worst case, where the condition of exponential weight increase *has to be* fulfilled, is where a single user shares traffic with all other users. If we transfer such results to the WCA problem, where there is no direct notion of link sharing available, we cannot know whether there is a corresponding situation. If there is no link sharing at all, any OOWA will be maximized by the BFC algorithm. So we also consider a *linear OOWA* with $w_i > w_{i-1}$ for any $i > 1$ in the following. In case that its maximum coincides with the maximum for the exponential OOWA, we can see this as an indication that such a virtual "link sharing" does not occur, or only to a small degree with low influence on related results.

Another special case for representing low link sharing will be newly introduced here. If a link never shares traffic with *more than one other user*, the requirement of exponential increase of the weights can be relaxed to the requirement $w_i > w_{i-1} + w_{i-2}$ for $i > 2$ and $w_2 > w_1$. One possible choice for such weights is $w_i = F_{i+2} - 1$, where F_i is the i-th element of the FIBONACCI series ($F_1 = 1, F_2 = 1$ and for $i > 3$ $F_i = F_{i-1} + F_{i-2}$). From $w_{i-1} + w_{i-2} = F_{i+1} + F_i - 2 = F_{i+2} - 2 = w_i - 1 < w_i$ it can be seen that this choice fulfills the relaxed weight increase requirement. We will call an OOWA computed with these weights a Fibonacci OOWA. The exposition can be extended by using more than two weights in a straightforward manner. We consider such special versions of the OOWA as a convenient tool for probing implicit sharing issues in any domain, where these expressions can be computed.

B. Application to WCA

In this section, we want to study the concept of exponential OOWA maximization for the WCA problem. We will consider two essential properties of an OOWA, and also study the relation between various forms of fairness. This will give the base for defining a suitable heuristic approach.

C. Some properties of the OOWA operator

The exponential OOWA comes out to be a "good" representation of fairness, as it numerically increases when values are replaced by their averages. But at first we recall

Lemma 1. *Among all OWA values with the same set of n data t_i in same order and weights $w_{(i)}$ in any order ($i = 1, \dots, n$), the OOWA takes the smallest value.*

For the rather simple proof, see [7]. Based on this, we note two ways to improve an OOWA (independent of the weight growth rule).

Theorem 2. *Given a set of data t_i and a set of weights w_i, each with n elements. If we select any subset of data and replace their values with the average of the selected data, the OOWA computed from the modified data set will never decrease.*

Proof: We consider the set of data t_i^* after modification, and sorted in decreasing order (see Fig. 1 for an example with 8 data points). Without loss of generality, we assume the weights to be already sorted in non-decreasing order, so $w_i = w_{(i)}$. The averaged data points will appear in a sequence within the ordering (darker green bars in Fig. 1). Now we modify them such that each average value becomes the value of the data point in the unmodified set of data, and such that the unmodified values become sorted in non-increasing order. Since the sum of all changes (decreases and increases) has to be 0, there will be a limit weight w_a so that in the sequence of average values, values with an index equal to or less than a will not decrease, and will not increase for values with an index larger than a (like index 4 in the example). If the sum of all increases of t_i values against the t_i^* values is S, then S will also be the sum of all decreases. Then it is clear that the weighted sum of all increases, where the weights are assigned by the order of the t_i^*, will not be larger than $w_a S$. Correspondingly, the weighted sum of all decreases, also weighted by the order of the t_i^*, will not be smaller than $w_a S$. Thus, when replacing the averaged values with the sorted unmodified values, while keeping the weight order, the value of the modified OOWA expression (it is an OWA now for some order of the weights, since the values are not sorted) will not increase. Now we still have to sort the data points into the correct non-increasing order to get the OOWA expression for the unmodified values. By Lemma 1, this will also never increase the OWA value. So we can transform the OOWA expression for the modified data set into the OOWA expression of the unmodified one without ever increasing the OOWA value. ■

Theorem 3. *Given two sets of n data points $t = (t_i)$ and $s = (s_i)$. If t_i Pareto-dominates s_i, then for any set of weights $w = (w_i)$ $OOWA_w(t) > OOWA_w(s)$.*

We can skip the proof here for space reasons, as the principle is the same as for Theorem 2.

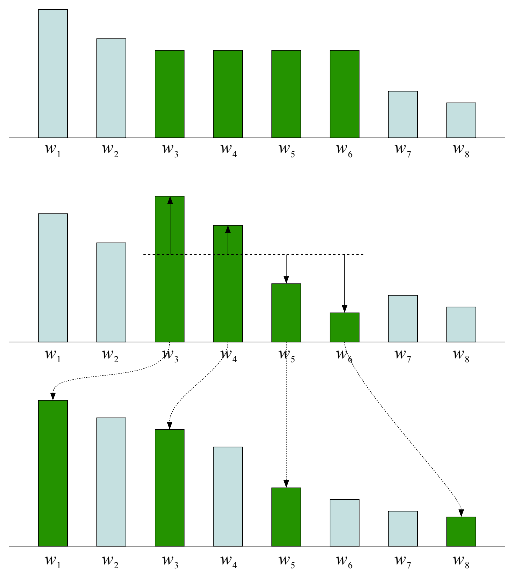

Figure 1. On the proof that exponential OOWA does not decrease when replacing any number of data values by their average. Note that $w_1 < w_2 < \cdots < w_8$.

D. *Feasibility of the exponential OOWA*

As the exposition in this paper is focussing on the exponential OOWA as a means to achieve fairness in the WCA problem, we want to explore the position of this fairness among the various other ways of representing fairness. For instances of the WCA problem, where the feasible space can be completely handled (in face of the exponential growth of the feasible space), we have derived several properties from the best element or maximum sets of fairness relations and the maximum value of some OOWA expressions. For each of these cases, the values of the properties were averaged over 1000 random problem instances, with channel coefficients to be chosen as uniform random numbers from $[0, 1]$. The results are shown in Table I.

The notations in Table I are: n_u - number of users, n_c - number of cells, M_{mmf} - maximum set for the maximum fair dominance relation, T_{eo} - maximum exponential OOWA element, T_{lm} - best element of lexmin relation, T_{lo} - maximum linear OOWA element, T_{fo} - maximum Fibonacci OOWA element, M_{pf} - maximum set of proportional fair dominance relation. A hyphen indicates that the value was not evaluated, as the property is always fulfilled (e.g. the expressions for linear and exponential OOWA coincide for $n_u = 2$). A star next to the number of cells indicates that for the average, due to larger computational effort, only 100 samples were taken. So, the last line for each number of users is for verifying trends. For handling proportional fairness, all performances containing 0 components had to be excluded.

First at all, while the definitions for fairness coincide in case of the network routing traffic allocation by the BFC algorithm, in case of the WCA problem they obviously refer now to different things. Nevertheless, some relations and trends are visible. We can observe (with the term "element" reffering to an element of the feasible space, i.e. the allocation of users to cells, and its performance vector):

- The size of the maximum set of the maxmin fair dominance relation is bounded from above by the number of users. For the case of a set of vectors with no pairwise equal component, this has been shown in [5]. However, no such bound is known for the proportional fair dominance relation so far, but the 7th column of Table I indicates that also here, the maximum sets are rather small (while definitely not bounded by the number of users, see result for 3 users and 5 cells).
- The maximum set M_{mmf} often contains the maximum exponential OOWA element, while this ratio decreases with increasing number of cells. However, a subsequent study (omitted here for space reasons) indicates that T_{eo} nevertheless, stays close to this set, and especially always closer than the best lexmin relation element T_{lm}. So we can confirm a tendency of the exponential OOWA to represent or approximate the maximum set of maxmin fair dominance relation, and better than the lexmin relation.
- Nevertheless, the maximal exponential OOWA element and lexmin element often coincide already, with a stronger decay of this ratio with increasing number of cells. Also, we can see that for a small number of users, we have to face issues where the maximum exponential OOWA element contains 0, i.e. users are virtually excluded. The same problem is known for maxmin fairness (see [6]), and the only thing we can say here is that this problem rapidly vanishes with increasing problem dimensions.
- There are, in general, only few cases, where the different OOWA values differ, in the majority of cases in more than 80% of the problem instances, there is no difference between linear, Fibonacci, and exponential OOWA. This is an indication that "link sharing" is not pre-dominant in the WCA problem, i.e. there is only a lower number of cases where the larger performance for one user needs to be compensated by *many* other users.
- Next to maxmin fair dominance, the exponential OOWA also seems to represent the maximum set of the proportional fair dominance relation. We can see that M_{pf} quite often contains T_{eo}, and this ratio does not seem to be strongly affected by increasing problem size. On the other hand, the best lexmin element is contained in M_{pf} to a lesser degree, and this ratio is much stronger falling with increase of problem size. The advantage of the OOWA here is that numerical values

Table I
AVERAGE VALUES FOR SEVERAL PROPERTIES OF FAIRNESS RELATIONS DERIVED FROM EXACT SOLUTIONS OF WCA.

n_u	n_c	$\lvert M_{mmf}\rvert$	$T_{eo} \in M_{mmf}$	$T_{lm} = T_{eo}$	$T_{lo} = T_{eo}$	$0 \in T_{eo}$	$T_{fo} = T_{eo}$	$\lvert M_{pf}\rvert$	$T_{eo} \in M_{pf}$	$T_{lm} \in M_{pf}$
2	2	1.845	0.848	0.805	-	0.137	-	1.185	0.863	0.969
2	3	1.847	0.679	0.501	-	0.034	-	1.687	0.946	0.688
2	6	1.709	0.679	0.590	-	0	-	1.586	0.874	0.509
2	8	1.680	0.645	0.555	-	0	-	1.669	0.783	0.415
2	9*	1.67	0.61	0.52	-	0	-	1.63	0.70	0.34
3	3	2.676	0.886	0.815	0.916	0.036	-	1.436	0.951	0.892
3	5	2.868	0.657	0.450	0.814	0.001	-	3.063	0.920	0.482
3	6*	2.27	0.73	0.65	0.87	0	-	2.86	0.87	0.56
4	4	3.161	0.914	0.811	0.882	0.005	0.977	1.640	0.953	0.828
4	5*	3.39	0.80	0.67	0.76	0	0.98	4.06	0.79	0.52
5	5*	3.50	0.93	0.84	0.85	0	0.96	1.58	0.91	0.80

of *all* performances are involved in its computation, while maxmin fairness as well as lexmin only refer to minimum values.

In summary it can be said that also for the WCA problem, maximizing the exponential OOWA of the performances among all possible allocations of users to cells is a reasonable way to represent fairness, as it comes close to other ways of defining fairness, despite some few weak points.

III. EXPERIMENTAL VALIDATION

The considerations of the former section will be summarized now to define a heuristic approach for approximating the maximum exponential OOWA element for a given instance of the WCA problem (i.e. a specification of channel coefficients for all cells for each user). We take advantage of Theorems 2 and 3 to define two local operators with the tendency to increase the exponential OOWA values.

1) We will use a *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) For the *swap* operator, we select two cells at random, and swap their user allocation. If this increases the performances for both users, the *swap* will be accepted.

Operator 1 refers to the fact that any OOWA will become larger or stay equal, if a set of numbers is replaced by their average. As we cannot manipulate performance values directly, since their values depend on sub-sums of channel coefficients, we want to enforce at least changes that make performances values at least more similar. Thus, the *replace* operator is not guaranteed to always increase the exponential OOWA, but may tend to do so. On the other hand, operator 2 refers to Theorem 3, and here, if the *swap* increases both performances, the exponential OOWA will increase as well.

In addition, we use an *annealing* probability p for both operators, where the operator is accepted with probability p even if it fails to fulfill the required property. The heuristic algorithm then itself is rather simple: start from a random allocation of users to cells, and apply operators 1 and 2 iteratively for a fixed number of times. Keep the maximum value of the exponential OOWA achieved over all steps.

The validation of such a heuristic can be only reasonable in cases where there is knowledge of the exact maximum value of the exponential OOWA. As mentioned before, the feasible space of the WCA problem grows exponentially, and up to our knowledge, there is no exact algorithm provided (yet) to find the maximum value analytically. Therefore, we use a rather small problem instance to compare the results of the heuristic approach with the known exact result. Here we report about results for a WCA problem with 4 users and 4 cells. The complete feasible space is small with a size $4^4 = 256$ but we are interested in the validation. For 1000 random problem instances, we have recorded the number of steps that are needed (one "step" here is application of one *replace* and one *swap* operator) to reach the maximum exponential OOWA by at least 99% (due to discrete nature of the problem, this usually means an exact match).

The result can be seen in Fig. 2, and it has been compared to some variations of the proposed heuristic. The plot shows the distribution of the average number of steps to reach the maximum value of the exponential OOWA of all possible allocations. Cases where more than 300 steps were needed (which can be seen as "failures") are included in the largest bin 300. We compared the - after some experiments - best choice of the annealing probability 0.2, with the case of using no annealing at all, the case of using a decay factor for the annealing, and the case of using annealing, but no *swap* operator. It can be seen that annealing is necessary to reduce the number of failures to an acceptable value (less than 1%). Without annealing, the maximum value is reduced fastest, but the number of failures is unacceptable high (not visible in the plot, the value is 653). Annealing

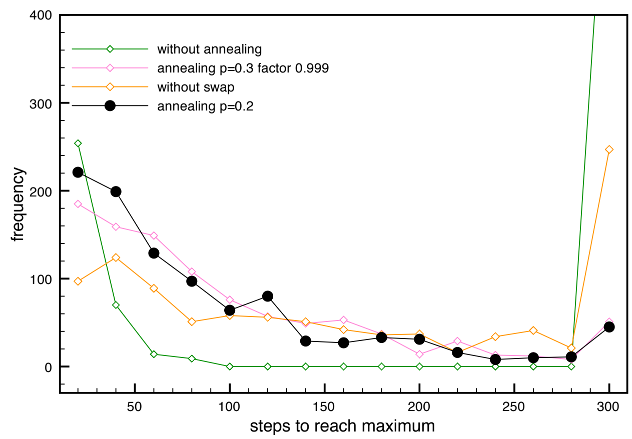

Figure 2. Validation of the proposed heuristic for a problem with 4 users and 4 cells, where the exact solution is known. Shown is the distribution of the number of steps needed to be closer than 1% from the true maximum of the exponential OOWA for 1000 random problem instances.

slows down the approach, but still needs fewer steps than the size of the feasible space in the majority of cases, with even rapid approach (less than 20 steps) for about 25% of the random instances. Using factored annealing, there is no notable difference, but skipping the *swap* operator has rather negative influence: the mean of the number of steps shifts to a larger value, and the number of failures increases strongly.

Thus, we can consider the choice $p = 0.2$ a reasonable approach to approximate the maximum value. Testing the same annealing heuristic on the next larger instance 5 users and 5 cells for 100 problems (search space size $5^5 = 3125$), and using bins of size 100, we can find the following frequencies: 10, 22, 20, 9, 6, 6, 1, 3, 9, and 13 failures, so an average of 150 steps and about 15% failures.

IV. Summary

In this paper, we have extended the concept of fairness in traffic congestion avoidance to the wireless channel allocation (WCA) problem. The main characteristics of the state achieved by the Bottleneck Flow Control algorithm, giving the base for the definition of fairness, were directly applied to the WCA problem, including lexmin fairness, maxmin fairness, proportional fairness, and, as a new result, exponential Ordered-Ordered Weighted Averaging (OOWA) operator maximization. A complete study for the case of small problem instances was provided, demonstrating the utility of exponential OOWA maximization for reflecting fairness in the WCA as well. Based on this and two basic properties of the general OOWA, a *replace* and *swap* operation based annealing heuristic was proposed, and it was studied on small problem instances as well, in order to justify their applicability to real-world instances of the WCA problem. Future work will focus on the improvement of such a heuristic, including a possible embedding into an meta-heuristic, or memetic search approach in order to expand the search capabilities.

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] 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.

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

[3] M. Köppen, R. Verschae, K. Yoshida, and M. Tsuru, "Heuristic maxmin fairness for the wireless channel allocation problem," in *Proc. 2010 International Conference on Broadband, Wireless Computing, Communication and Applications (BWCCA10)*, 2010, pp. 606 –611.

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

[5] 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.

[6] M. Köppen, K. Yoshida, and M. Tsuru, "Fuzzification of maxmin fairness relation based on subvector dominance degree," in *Proc. 2010 International Conference on Intelligent Networking and Collaborative Systems (INCoS 2010)*, 2010, pp. 344–349.

[7] M. Köppen, A. Honda, and J. Okamoto, "Fuzzy fusion fairness relations for the evaluation of user preference," in *Proc. 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011)*, 2011.