Modeling the Over-the-top flow control optimization in multi-provider wireless heterogeneous networks

***Abstract*—Resource allocation optimization is critical to system-wise performance of a wireless heterogeneous networks (HetNet). Flow control optimization problem is one type of resource allocation problem, which tries to reach certain system-wise optimization guidelines by controlling the flow association and rates at the ends of communication sessions in a wireless HetNet. The previous literature mainly focuses on single-provider HetNet, and failed to provide an in-depth evaluation to several fundamental problems in the system design of this new type of optimization problem, i.e. 1) the distances of various association schemes to the optimality; 2) the sources and impact of potential throughput estimation error; 3) the modeling and performance difference when multiple interfaces can be used simultaneously. Previous literature mainly assumes single interface with rough association based control granularity. In this paper, we first model the system with a similar assumption as the previous literature, and study the first two problems. We then further generalize the model to reflect the latest network stack development reality, where multiple interfaces can be used at the same time with various granularity. By simulations, we found interesting properties of the flow control optimization problem under various assumptions. This provides theoretical foundation and parameter selection guidance to future system design and implementation of this type of flow control system in wireless heterogeneous networks.**

I. Introduction

Mobile data is growing at a rapid rate. The emergence of Machine-to-Machine (M2M) and Internet-of-Things (IoT) applications will add further traffic to the existing and sometimes congested wireless networks. The Federal Communications Commission (FCC) is aware of this emerging 'data tsunami' and has called for technical and business innovations to increase the efficiency of spectrum usage [1]. Even though more installations and upgrades of base stations can help to solve these problems, integrating and careful planning the usage of existing network resources remains the most cost-effective avenue to overcome the challenges and issues. Generally speaking, we call this planning of the usage of the limited network resource for better system-wise network performance, the **resource allocation** (RA) problem. Specifically, we are inertested in resource allocation problem in wireless systems.

From an abstract perspective, a wireless system, like shown in Fig. 1, consists of three components,

1) Communication ends - at least one end is wireless.
2) APs - can be of the same or different RATs and operators.
3) Backhaul nodes - routers, etc.

The most general form of the resource allocation problem for a wireless system can control the resource allocation/usage

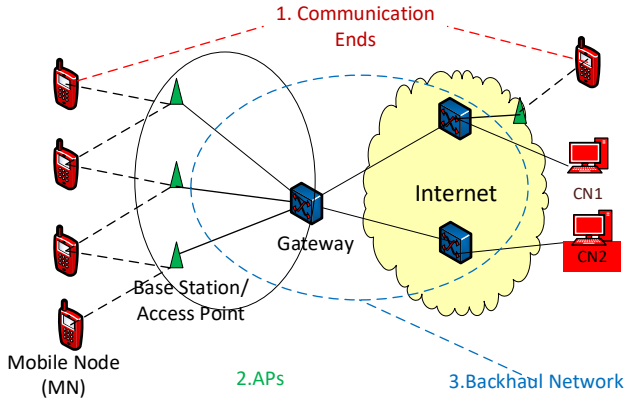


Fig. 1. System context.

planning at all of the nodes of the above three components with any granularity. The controls at the communication ends can involve controlling user associations and data sending rates at the end-point client devices, like in [2]. The controls at APs can include the innovations of the scheduling algorithms at single APs, like in [3]. The control at backhaul network can include the resource allocation and flow scheduling study at the backhaul routers [4].

We first try to provide a generalized mathematical formulation of the above three resource allocation problems. Network sessions usually have uplink and dowlink. To simplify the discussion here, we first think about one direction, e.g. downlink. In Fig. 2, we abstract the system in Fig. 1 into a graph. For any internal node in this graph, it has one or more inputs and outputs. For any node i, we denote its input nodes as $S_i = \{s_1, ..., s_s, ..., s_m\}$ and the nodes in the output direction as $T_i = \{t_1, ..., t_t, ..., t_n\}$. For the most downstream nodes with no outputs, they represents the end users in a real system. We denote the set as U, and each of them as **u**. We denote the size of the flow (throughput) from any node j to i as $d < j, i >$.

Despite the details, resource allocation in any network node will result in resource division and then a vector of resulted throughput. The constraint $d < i, t >= f(c < s, i >, d < s, i >)$ means exactly this, where s means a source node, and t represents a destination node as shown in Fig. 2. $d < s, i >$ can also be considered as the output throughput when node s is considered as the current node i. The function f abstracts the mapping from the inputs to the resulted throughput outputs. To clarify what the function f represents, we can think f as two functions. $r < i, t >= p(c < s, i >, d < s, i >)$, where $r < i, t >$ means the actual portion of resource allocated to

output destination t. The second function $d < i, t >= q(r < i, t >, c < s, i >, d < s, i >)$.

Also, we denote the sum of flow input throughput of node u as $D_u = \sum_{s \in S_u} d < s, u >$. The optimization objective is to maximize an utility function of the vector of D_u, denoted as $\hat{D_u}$. The second constraint means the sum of the out throughput is less or equal to the capacity of node i (C_i).

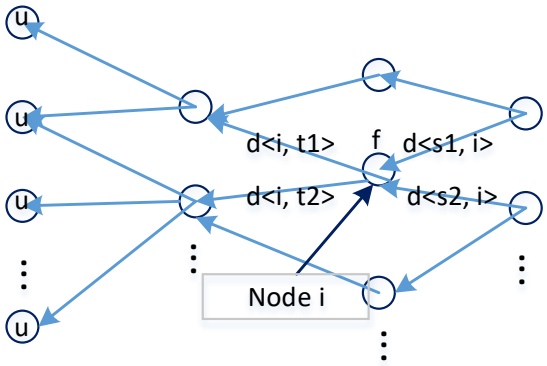


Fig. 2. System context.

$$\text{Maximize} \quad \sum_{u \in U, s \in S_u} U(\ d < s, u >)$$

subject to

$$d < i, t >= f(c < s, i >, d < s, i >), \quad (1)$$
$$\forall i \in V, t \in T_i, s \in S_i$$
$$\sum_t d < i, t > \leq C_i.$$

We notice that this is similar to the model in some ATM network resource allocation work []. However, in a heterogeneous wireless system, there are many limitations that can add constraints to the above problem. For example, the heterogeneous demands and underlying network protocols. Most of the previous research on resource allocation assumes uniform backlogged traffic. This is not realistic in real networks. Meanwhile the ATM alike resource allocation schemes requires homogeneous underlying network protocol, which has proven to be unrealistic in an IP world. Therefore, this paper mainly focuses on the first optimization route, i.e. only controlling the associations and rates at the communication ends. We call it Endpoint Flow Control Optimization (EFCO).

At the same time, heterogeneity is becoming a major characteristics in wireless networks nowadays. One location is usually covered by cellular networks of different operators, and WiFi APs of different owners. Fig. 3 illustrates a typical multi-provider wireless HetNet environment. The HetNet consists of multiple sub-networks of different providers covering the same area. For example, we can think of the larger sub-network (green) as an LTE network, while the smaller ones (orange) as WiFi networks. Each mobile device has at least one interface that can connect to each of the sub-network. Its traffic can choose to go through either or both of the

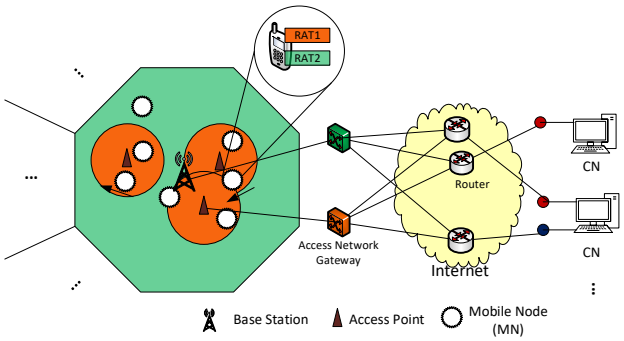


Fig. 3. System context.

interfaces. If multiple overlapped sub-network of the same RAT from different providers coexists, the mobile device can further choose which one it associates to. Every mobile devices connects to one or more corresponding nodes (CN) via Internet. We focus on the optimization of downlink traffic like video streaming from CNs to the mobile nodes, as this type of traffic takes up most of the Internet traffic. (The dotted connection in this figure means there can be more internal nodes in that path.)

There are plenty of previous literature on resource allocation problems in wireless HetNets. From the top level, we categorize it into Over-The-Top (OTT) and non-OTT solutions. This categorization is based on whether the solution relies on the modification of the internal resource allocation schemes, like the resource allocation algorithms or queuing disciplines at the routers or APs inside a wireless network. An OTT solution, for example, only relies on the flow control at both ends (usually a mobile node and a remote server) of communication sessions to indirectly control the resource allocation. Examples of OTT solutions include [2], [5], [6], [7], [8], while [3], [9], [10] are all of non-OTT. Since the OTT solution only controls either the flow association or rates at the communication session ends, we can also call it end-point flow control. This paper deals with the OTT solutions only.

This paper focuses on how to model the flow control problems in a wireless HetNet under an OTT assumption. It explores the implications of various assumptions to the problem modeling, and tries to answer the following questions, 1) what are the distances of the flow control schemes in the previous literature to the optimal in terms of the flow control objective function value and other system performance metrics; 2) what are the potential errors in the previous schemes and their impact to the performance of the solution; 3) how to model the problem if flows can be split to multiple active interfaces with finer granularity.

The first question is why we want to focus on OTT solutions in a wireless HetNet. We observe that wireless HetNets are becoming more and more heterogeneous. One of the dimensions of this heterogeneous property is the ownership of the sub-networks. The newly developed network stacks and applications can use more than one interface at a time to send/receive data [11], [12]. The wireless networks those

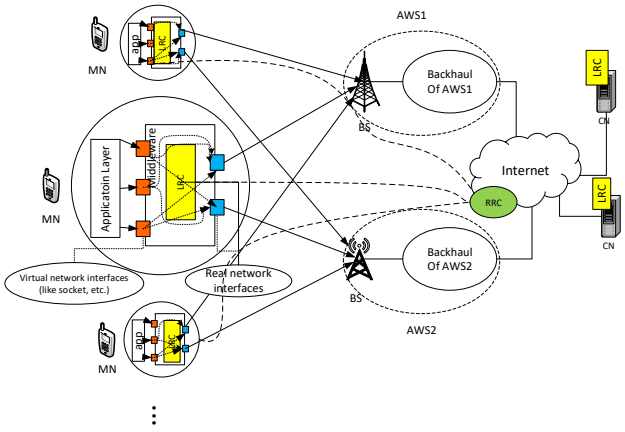


Fig. 4. A generalized OTT architecture for the flow control in MP-HetNet.

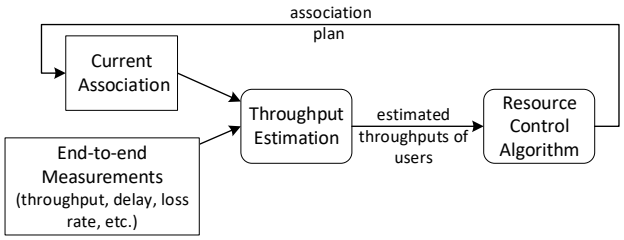


Fig. 5. Work flow of the system.

interfaces use typically do not belong to the same provider. For example, the LTE interface uses Sprint LTE network, while the WiFi interface connects to campus WiFi network. We call this type of HetNet a multi-provider HetNet (MP-HetNet). For the resource allocation in a MP-HetNet, a natural requirement is an OTT design. Because it is difficult to deploy the same internal scheduling/queuing discipline change, when the sub-networks are owned by different providers.

To meet the OTT requirement of the above MP-HetNet, we assume the following general conceptual system architecture and components in the solutions we discuss below. As shown in Fig. 4, a Local Resource Controller (LRC) locates at every end device, i.e. a User Equipment (UE) or a remote server. It collects network connection status information from the device and relay it to the Regional Resource Controller (RRC). The RRC then use the aggregated information to model the throughput of every client device under various planning choices. Centralized algorithm running at the RRC then produces an optimized solution for how every flow should be associated so that system-wise objective can be optimized. The LRC receives the plan and enforces it locally. Though we present the RRC as a centralized service, its storage and computation can be distributed. It is only a centralized service conceptually.

The abstracted data flow of the whole system is shown in Fig. 5. Clients first send flow information measured from the two ends of communication sessions to the RRC. The RRC then conducts resource allocation algorithm which invokes the throughput estimation. This is because the scheduling algorithm needs to know the estimated throughput to judge which association policy is better. The figure shows the relation of different modules of the whole designed system, and why throughput estimation module is needed and important for the whole system.

This paper is organized as the following. We go over the previous literature in section II. Then we present a generalized mathematical model of the exiting flow control schemes in section III.

In section IV, we first model the system with a similar assumption as the previous literature, and examine the performance of some existing flow control schemes in the previous literature. We would like to see their performance compared together with the optimal in a systematic and control manner. Under the single-active-interface assumption, in section V, we explore the causes of the throughput error estimation, and test the impact of the input errors from two perspectives to the performance of the flow control schemes. We first explore arbitrary error levels which can abstract any kine of error. Then, we specifically test the errors because of control frequency/delay. Finally, we extend the model in section VI to to multiple-active-interface, and finer flow control where client can use partial of the network capacity that is available to it, which solves the problem 4).

II. Related works

Though there are a lot of previous literature on HetNet resource allocation problem, and some of them even of OTT type, there is no good modeling work that tackles with some of the fundamental problems in such an OTT type of flow control systems.

First of all, none of the previous literature shows the distances of

It includes, the impact of 1) throughput estimation error; 2) the control frequency; 3) simultaneous usage of multiple interfaces; 4) granularity of flow control. Besides, there is no comparison of previous schemes in different performance metrics. Previous literature mainly assumes rough granularity of single interface with backlogged traffic.

III. Generalized mathematical model of the problem in the previous literature

We first model the single-active-interface usage scenario here similar to [2], [5], [6], [7],

$$\begin{aligned} \text{Maximize} \quad & \sum_{j=1\ldots M} \sum_{i=1\ldots N} log(\ T_{ij}) * x_{ij} \\ \text{subject to} \quad & \\ & \sum_{j} x_{ij} = 1, \\ & T_{ij} = \mho(\hat{x}, ...), \\ & x_{ij} \in \{0, 1\} \end{aligned} \tag{2}$$

The index i is for UE, while j for AP. N is the total of UE, while M is that for AP. x_{ij} means whether the user i should connect to AP j or not. It should be either 0 or 1. The solution

to the above problem generates a set of x_{ij}, which we denote as $\hat{x}$. $\hat{x}$ is the flow association guideline provided to UEs from the RRC. The first constraint means every UE can only connect to one interface in this first model that is similar to the previous literature. T_{ij} is the end-to-end real throughput of UE i if it is connected to AP j. The second constraint shows that the end-to-end throughput is a function that involves the association plan $\hat{x}$, and other factors. Those factors can include the signal-to-noise-ratio of the concerning link and the characteristics of the other internal nodes in the path, like loss rate of a router in the path. Note that this function $\mho$ abstracts the individual scheduling scheme used by every AP. It is similar to the model in Eq. (1) - (4) in [2], but with more generalized form. The overall optimization objective is global PF as in the other previous literature listed above. We can see this is a generalized model for the single-active-interface scenario. We will explore the model of multiple-active-interface scenario in section VI.

IV. Examine the performance of the existing flow control schemes

Given the above optimization problem in Eq. 2, the first question is, if all the scheduling algorithms had a perfect input, i.e. 100% accurate throughput estimation (as in Fig. 5), what are the distances of the current flow association algorithms to the optimal solution. Most of the previous literature fails to provide a detailed comparison to the optimal solution. Also there is no existing report of the comparison among the schemes. In this section, we first pick several representative flow control schemes used in the real systems and proposed in the previous literature, and compare them in a conceptual and more controlled manner.

First, we set up a scenario that has N UEs and M APs. To make the percentage of UEs with good connection status to each AP more controllable, we did not use random mobility. Instead, we control how much percentage of users connect to each AP. For the type of APs, we test with two cases:

Case 1. *Every AP uses proportional fairness as its scheduling scheme.* (This is similar to the assumption in [2], [5], abbreviated as 'pf only' in figures).

Case 2. *Some AP uses proportional fairness, while others uses throughput fairness (like a WiFi AP), abbreviated as 'both' in figures* (like in [6]).

This assumption actually has significant implication to the modeling. With the former, the objective function is concave and the whole problem is convex after the logarithm function. Therefore, we can use convex solvers (e.g. the interior point method solver in MATLAB) to solve the problem easily. However, with the added complexity of throughput fairness scheduling APs, like in [6], [13], the objective function will not be concave anymore, and it will not be feasible to solve the optimal using convex solvers. We will show the impact of this to the convex solver based method (like the first centralized method in [5]) in the result part. In each scenario, we assume there is one AP with larger coverage, like a LTE macrocell; while the rest are with a small coverage like a WiFi AP or a picocell. We control the percentage of users under the smaller coverage APs (P_s). We test with P_s = {0.8, 0.6, 0.4, 0.2}. We use the distance to throughput mapping we measured from NS3 for the rates.

We compare the following five methods.

1) policy-based: This represents the interface selection scheme on most of smart phones. Whenever WiFi is available, it tries to use WiFi. If WiFi not available, then it connects to the cellular network. Among all the WiFi APs, it selects the interface with the best signal strength related metric (like SNR, RSRQ, etc.) In our simplified scenario, where SNR is a function that is proportional to $\frac{1}{d^2}$, where d is the distance to an AP, the algorithm just pick the nearest small-coverage AP.
2) optimal-brute-force: This generates the optimal solution using a brute-force method which iterate through all the possible flow configurations.
3) ATOM: The method in [6].
4) random: It selects 5 randomly generated flow configurations, and returns the one with the best objective function value.
5) round-off-interior-point: This uses a non-linear solver (*fmincon* in MATLAB) to solve for the problem.

We compare the following three metrics,

1) PF value : This is simply the objective function value in Eq. 2.

2) Aggregated throughput : This is the sum of all the user throughputs

3) Jain's fairness metric :
$J(T_1, T_2, ..., T_n) = \frac{(\sum_{i=1...n} T_i)^2}{n * \sum_{i=1...n} T_i^2}$

We test with N = {5, 10} UEs and M = 3 (i.e. one large coverage AP and two small coverage APs). For every time slot, we generate the distance to each AP randomly following the coverage percentage P_s.

Figs. 6 and 7 show the results for the Case 1, i.e. only PF scheduling APs, while Figs. 6 and 7 show the results for the Case 2, i.e. both PF scheduling APs and throughput fairness APs.

We have the following observations from these figures,

1) optimal-brute-force always has the best PF value, which serves as the baseline and a sanity check.
2) round-off has close-to-optimal performance under Case 1 most of time. However, in some rare cases the round-off can make way worse configuration that results in only about half the optimal PF value and aggregated throughput, which are even worse than those of the policy based and random. This can be clearly observed in Fig. 30(b).
3) ATOM is closer to the optimal, and with lower standard deviation compared with random and policy-based.
4) Optimal-brute-force wins with larger margin under Case 2 compared with Case 1 in terms of aggregated throughput. This is reasonable because the objective function in Case

5) The methods have a more mixed Jain's fairness index performance under the Case 2.
6) Random method has the worst performance and largest deviation most of time. The policy based method is only slightly better than the random generated configurations.
7) ATOM has larger standard deviation in terms of both aggregated throughput and Jain's fairness index, compared with the optimal, even through they have similar PF values.
8) The optimal has larger standard deviation in terms of Jain's fairness index under Case 2 compared with Case 1.
9) For N=5, M=3, the random method has better throughput sometimes, but when N=10, M=3, the random method is the worst. This is because we only try 5 random solutions no matter the number of UEs. We believe this is a reasonable setup, because in practice, a real-time random solution can not scale up with the UE number, as the UE number can be very large.
10) From the additional results in the appendix, we can see round-off-interior-point is better when WiFi coverage rate is low (P_s=0.2 and P_s=0.4), at least in the scenario of N=10, M=3, and Case 2 APs. For Case 1, it can get solution very close to optimal. This proves the result from the [5] is correct. However, when extending the HetNet to a more general HetNet which does not only contain PF scheduler APs, the performance of this method degrade dramatically.

The result in this section shows, even with perfect information, the current algorithms will have some distance to the optimal global PF objective. This has very important implication for the study below about the impact of the input error to the algorithms.

So, when estimating the impact of input error in the section V, we need to start with the optimal-brute-force solution. Because only the optimal solution do not have the confusion that where the performance difference comes from.

V. Throughput Estimation Accuracy

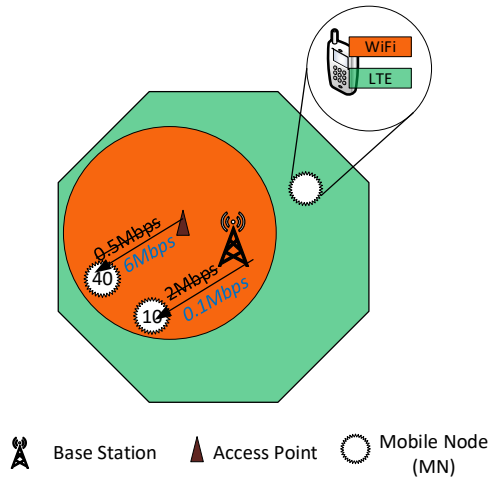


Fig. 10. Simple example showing why the throughput estimation accuracy matters.

We first use the following simple example to demonstrate the importance of the throughput estimation accuracy to the performance of the whole flow control optimization system. Like shown in Fig. 10, there are one LTE BS and one WiFi AP each with a capacity of 20Mbps. 10 users with an estimation of 2Mbps throughput all connected to AP 1, but actually every of them only uses 0.1Mbs. The neighboring AP2 has 40 users with an estimated throughput of 0.5M and an real demand of 6Mbps per user. As we can see, each of them can only reach 0.5Mbps. The scheduler thought the overall throughput is 20M + 20M = 40M, but actually it is only 1M + 20M = 21M. Moving some of the 6Mbps users to the first AP can definitely better balance the system and achieve better proportional fairness utility, but scheduler fails to do so because the input throughput estimation errors. From the above results, at least intuitively the throughput estimation accuracy matters for the outcome of overall throughput. In this section, we will try to inspect the impact of it to various scheduling algorithms in details.

Readers may have the question of where the errors come from. In general, the throughput estimation errors can come from,

1) throughput model, which includes errors from,
   a) capacity modeling error;
   b) individual AP resource sharing modeling error;
   c) failing to include flow demands in the model;
   d) failing to include end-to-end effects in the model.
2) the gap between sampling/prediction of the inputs (like the SNR) and their change due to system dynamics such as fading and mobility
3) the dynamics of control delay

The control delay for a client device in 3) is defined as the time from the input measurement (like SNR) are sampled at a device to the time the flow control policy is received and enforced at the device. The dynamics of control delay can result from control packet loss or delay. Also, we note that 2) and 3) are closely related. Because if only sampling (no prediction) is used, the functionality of the designed system will highly rely on low control delay and system dynamics. If prediction is used, considering the control delay is critical for a prediction to the dynamic metrics of the next time slot.

From a systematic perspective, as shown in Fig. 5, an appropriate throughput estimation model is essential to the design and functionality of the whole system. This is because it provides input to the flow control algorithm. However, the previous literature never studys the sensitivity of the flow control algorithms. Most of them simply assume the throughput model has a 100% accuracy. However, as we can see in subsection V-A, this assumption can not hold in real wireless network. So, we study the sensitivity of some representative flow control algorithms to various levels of input error rates. We would like to select the same set of algorithms as in section IV. However, the problem is invalid for the policy based and the random methods by the nature of the algorithms.

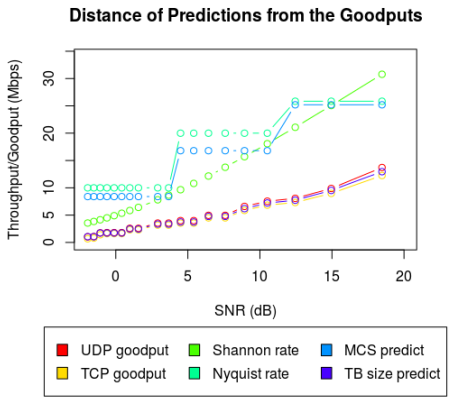


Fig. 11. Distance of estimation methods to the real throughputs.

Thus, we only compare three algorithms in this section, i.e. 1) brute-force 2) ATOM 3) round-off-interior-point.

The first subsection below (subsection V-A) first deals with the first type of error, and the next subsection V-A2 tries to build simplified model to test the impact from 2) and 3).

A. Type I error

1) Preliminary Experiments to show the errors exist: It might not be clear to some readers why the error exists. We use the following simulation in NS3 to show why the rough throughput model used in previous literature can have significant error introduced to the throughput estimation.

We tested the error rates when using various methods to estimate user throughput with the LTE module in NS3. We test with the following methods, and compare with the TCP and UDP goodput,

1) Shannon Equation

$$T = B * log(1 + SNR)$$

2) Nyquist Equation

$$2B * \log_2(Nbits)$$

Where Nbits is the number of bits used for the coding scheme.

3) Modulation and Coding Scheme (MCS) predict For example, if it is useing 64QAM, every symbol uses 8bits. In LTE, a symbols can be supported for a 10MHz channel. The prediction result will be $8a$ bps in this case.
4) Transport Block (TB) size The notion of transport block is a physical layer rate limiter in LTE. It is used to control the error rates in the physical layer. Basically, the sets up uppper bound for the maximum number of bytes one UE should send or receive given specific MCS and number of resource blocks. The talbe of TB size can be found in 3GPP standard TS36.101, Annex A.2.1.2 [14].

We tested with two cases, i.e. 1) no fading 2) a fading model defined in [15] based on 3GPP fading propagation conditions (see Annex B.2 of [16]). We used pedestrian model with a speed of 3 kmph. Fig. 12 provides visualizations of the second fading model.

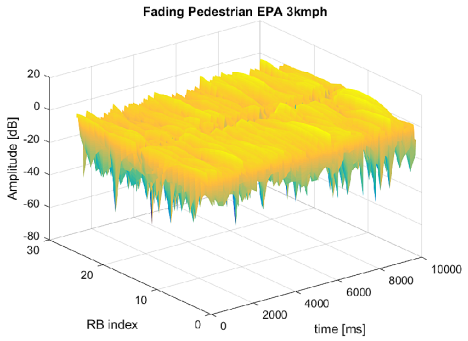


Fig. 12. NS3 fading model (pedestrian 3 kmph).

Results with no fading

Fig. 13 shows the best fitting of SNR based estimation curves to the TCP throughput. Fig. 14 shows the best fitting of SNR based estimation curves to the UDP throughput.

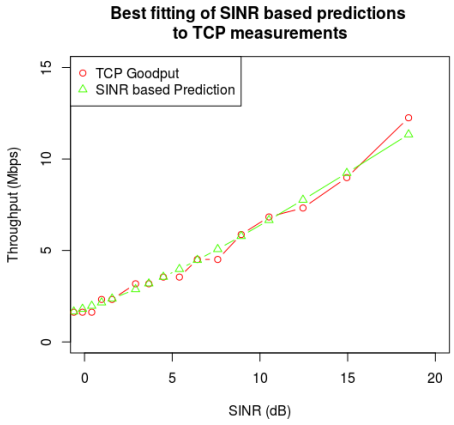


Fig. 13. The best fitting of SNR based estimation curves to the TCP throughput. (no fading)

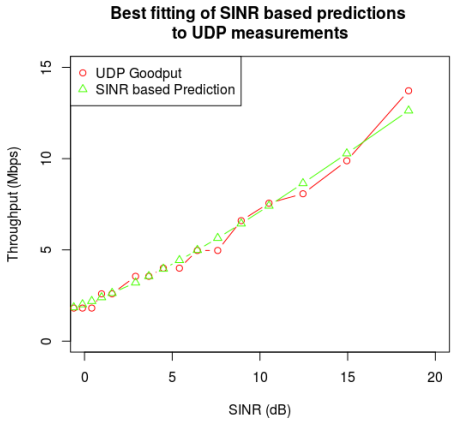


Fig. 14. The best fitting of SNR based estimation curves to the UDP throughput. (no fading)

Results with fading

Fig. 15 shows the best fitting of SNR based estimation curves to the TCP throughput. Fig. 16 shows best fitting of SNR based estimation curves to the UDP throughput.

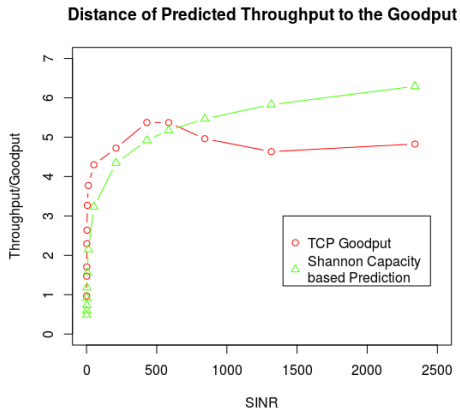


Fig. 15. The best fitting of SNR based estimation curves to the TCP throughput. (pedestrian 3kmph)

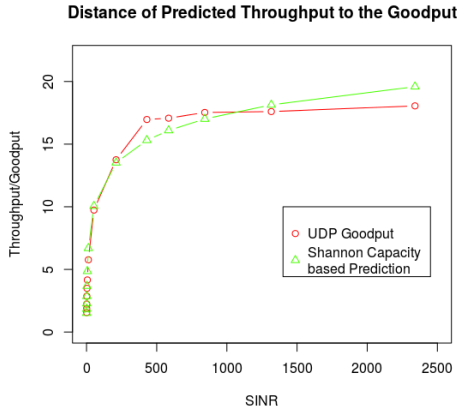


Fig. 16. The best fitting of SNR based estimation curves to the UDP throughput. (pedestrian 3 kmph)

From the results, we can see that the throughput estimation, even with the best fitting parameter, can be larger or smaller than the real throughput. And, in real system, there is no easy way to achieve a best fitting parameter. So, the throughput estimation errors exist using the previous methods. We can also see that the throughput estimation error is much larger when a more realistic fading model is used.

2) Conceptual sensitivity tests: We set up a similar scenario like in section IV. However, this time we purposely insert fixed error with random directions. The error rates we inserted e = $\{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$. The directions we chose is with half positive errors and half negative ones. Every experiment lasts for 10000 time slots. Figs. 18 to 20 show the results.

As we discussed in section IV, only the optimal-brute-force can clearly show the impact from only the purposely added errors. The other methods will also include the impact from the errors of the method itself. From the subfigures (a) of Figs. 18 to 20, we observe that,

1) From Fig. 18(a), we can see, for the optimal solution, the input error will always lead to degraded PF objective function value. Meanwhile, the performance degradation is nearly proportional to the error inserted.
2) From Fig. 19(a), we observe that the system performance in terms of aggregated throughput may both increase and decrease. The CDF of this metric is almost symmetric. This means in all the error levels (up to 50% error), the optimal-brute-force method will get almost equal chance to achieve a degraded throuhgput, as it can get a better one. Different from that, from Fig. 20(a), we see that the CDF of the Jain's fairness index can also increase and decrease, but biased to the right side of the x axis, which means the method has larger chance to get a better solution in terms of Jain's fairness metric.
This means after optimizing a scenario with input errors, the results can be better than the one without errors in terms of aggragated throuhgput and Jain's fairness index. This is different from the PF objective value. This is because the PF optimality does not gurantee the optimality of either aggregated throughput or Jain's fairness inedex. It is only a trade-off between the two with emphasis on the throughput. And, we can see that Jain's fainess can get a much better solution when input errors inserted. This means errors will introduce random divergence from the original optimization destination, which may end up with either improved or degraded solutions.
To observe the detailed impact of the input errors to the aggregated throughput and fairness respectively, we overlay the changes of both throughput and Jain's fairness metric onto the same figure. Making the figure readable, we sample the first 100 timeslots from the 10000 timeslot run. Fig. 17 shows the results of optimal-brute-force under various input error rates. For the results of the other methods, please refer to the appendix Section VIII-A of the the full version of the paper. In the figures, the x axis is the time sequence; while the y axis means the percentage of change of the two different metrics. If the aggregated throughput was positive, it is drawn in blue box; while grey box if negative. If Jain's fairness metric was positive, it is drawn in red box; while white box if negative.
3) From Figs. 19(b) and 19(c) we can see that the errors from the method themselves have made the results from optimizing the scenario with errors better with more than half cumulative possibility. It first sounds like a good thing. But, we can see that the worst case is with much larger degradation also. This means the results after two errors is not that predictable. They are only helpful from a possibility perspective.

B. Type II & III errors

We simulate and verify the impact of the Type II & II errors in the following way. We assume the system has an atomic time unit that SNR will change based on. Then, we vary the control frequency as the multiples of this time unit. The multiples we tries α = $\{1, 3, 7, 15, 31\}$. We continue to use N=5, M=3, and the WiFi coverage rate of P_s= 0.8. For the results with more P_s values, please refer to the appendix

at the full version of the paper. We test with SNR change rate (C_r)= { 0.1, 0.3, 0.5}.

$$C_r = \frac{SNR_t}{SNR_{t-1}}$$

where t is the any time slot.

Figs. 21 -23 shows the results of Case 1 APs, i.e., proportional fairness APs only. Figs. 24 -26 shows the results of Case 2 APs. From the results, we have the following observations,

1) For ATOM and optimal-brute-force, the impact of whether the scenario is using Case 1 or 2 APs is not large, compared with the round-off-int solution. As we can see from Figs. 23 and 26, the difference is much larger between this pair of results. This is because from the distance-to-optimality study in Section IV, we have seen that the error of round-off-int is much larger than those of the other two methods.
2) When the control time interval is larger than 10 times of the SNR change time interval (like 15 and 31 times), the optimization objective value starts to degrade. For the scenarios with a smaller control time interval, the performance degradation is negligible.
3) The performance degradation is also related to the amplitude of the SNR change. For example, when the SNR change rate is 0.1, the performance degradation of optimal-brute-force is very small. However, when the SNR change rate increased to 0.3 and 0.5, the degradation is obvious.

VI. Multiple-active-interface and Flow control granularity

Most of the previous OTT type flow control optimization literature [2], [5], [6] model the problem as following,

As indicated in [5], if the constraint of the association variable x must be from $\{0, 1\}$ can be removed, the relaxed problem will not be NP-hard anymore, and can be solved by non-linear solver. Both [5], [17] have mentioned this *fraction association* problem, and think it is unrealistic in real world. They eventually used the round-off integral solutions as the final policy. The question we want to ask is *what is the real physical meaning of a fractional association, and what does it mean to the throughput estimation model?*

VII. Conclusion

In this paper, we provides an in-detail analysis of how to model an over-the-top type resource allocation optimization system for a heterogeneous wireless systems. From the results, we can see that the performance of the designed system relies on system parameters like control message frequency, and can be largely impacted by input error. Different resource allocation algorithms behave differently to these impacts.

VIII. appendix

A. Various WiFi coverage rates for the distant to the optimal solution study

1. P_s **= 0.2**

Figs. 27 to 30 shows the distance-to-optimality study results when P_s = 0.2.

2. P_s **= 0.4**

???????? shows the distance-to-optimality study results when P_s = 0.4.

3. P_s **= 0.6**

???????? shows the distance-to-optimality study results when P_s = 0.6.

B. Various WiFi coverage rates for the input error study

References

[1] FCC, "Fccs national broadband plan," http://www.broadband.gov.

[2] T. Bu, L. Li, and R. Ramjee, "Generalized proportional fair scheduling in third generation wireless data networks," in *Proc. of INFOCOM*, April 2006, pp. 1–12.

[3] R. Amin, J. Martin, J. Deaton, L. DaSilva, A. Hussien, and A. Eltawil, "Balancing spectral efficiency, energy consumption, and fairness in future heterogeneous wireless systems with reconfigurable devices," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 5, pp. 969–980, May 2013.

[4] M. Shreedhar and G. Varghese, "Efficient fair queuing using deficit round-robin," *IEEE/ACM Transactions on networking*, vol. 4, no. 3, pp. 375–385, 1996.

[5] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. Andrews, "User association for load balancing in heterogeneous cellular networks," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 6, pp. 2706–2716, June 2013.

[6] R. Mahindra, H. Viswanathan, K. Sundaresan, M. Y. Arslan, and S. Rangarajan, "A practical traffic management system for integrated lte-wifi networks," in *Proc. of MobiCom*, ser. MobiCom '14. New York, NY, USA: ACM, 2014, pp. 189–200.

[7] S. Deb, K. Nagaraj, and V. Srinivasan, "Mota: Engineering an operator agnostic mobile service," in *Proc. of MobiCom*. New York, NY, USA: ACM, 2011, pp. 133–144.

[8] A. Sridharan, R. Sinha, R. Jana, B. Han, K. Ramakrishnan, N. Shankaranarayanan, and I. Broustis, "Multi-path tcp: Boosting fairness in cellular networks," in *Proc. of ICNP*, Oct 2014, pp. 275–280.

[9] H. Zhang, F. Bai, and X. Ju, "Heterogeneous vehicular wireless networking: A theoretical perspective," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*. IEEE, 2015, pp. 1936–1941.

[10] Y. Du and G. de Veciana, "Scheduling for cloud-based computing systems to support soft real-time applications," 2016.

[11] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "Rfc6824: Tcp extensions for multipath operation with multiple addresses," 2013.

[12] A. L. Ramaboli, O. E. Falowo, and A. H. Chan, "Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1674–1690, 2012.

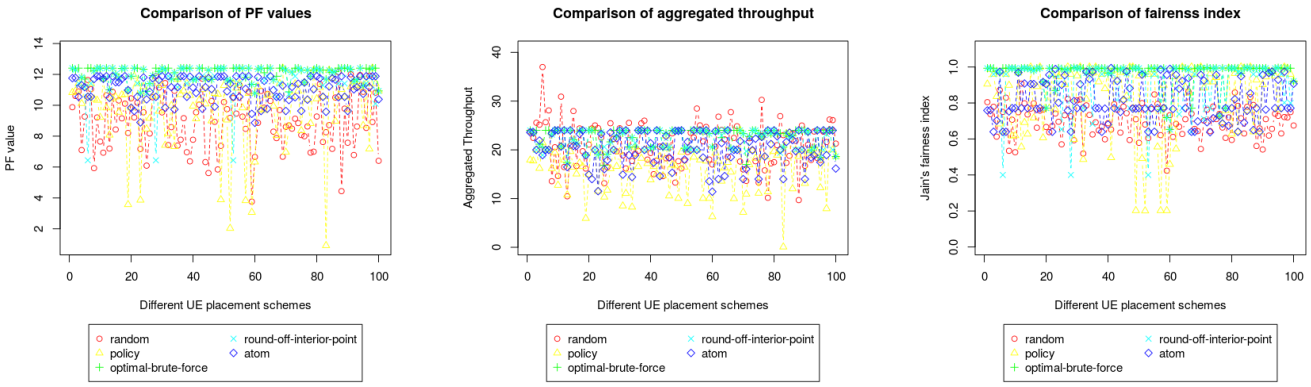
[13] W. Wang, X. Liu, J. Vicente, and P. Mohapatra, "Integration gain of heterogeneous wifi/wimax networks," *Mobile Computing, IEEE Transactions on*, vol. 10, no. 8, pp. 1131–1143, Aug 2011.

[14] E. U. T. Radio, "User equipment (ue) radio transmission and reception, 3gpp std. ts 36.101."

[15] G. Piro, N. Baldo, and M. Miozzo, "An lte module for the ns-3 network simulator," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2011, pp. 415–422.

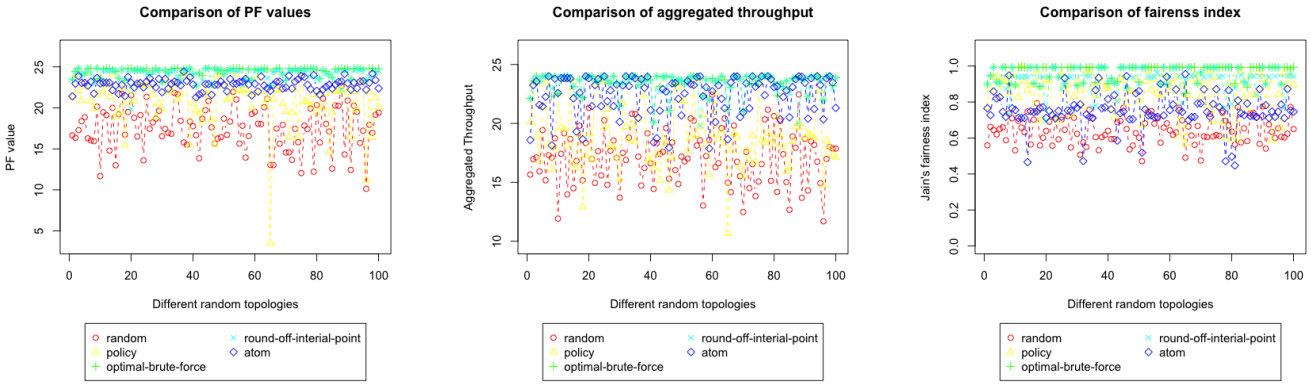
[16] E. LTE, "Evolved universal terrestrial radio access (e-utra); base station (bs) radio transmission and reception (3gpp ts 36.104 version 8.6. 0 release 8), july 2009," *ETSI TS*, vol. 136, no. 104, p. V8.

[17] R. Amin, "Towards viable large scale heterogeneous wireless networks," Ph.D. dissertation, Clemson University, 2013.



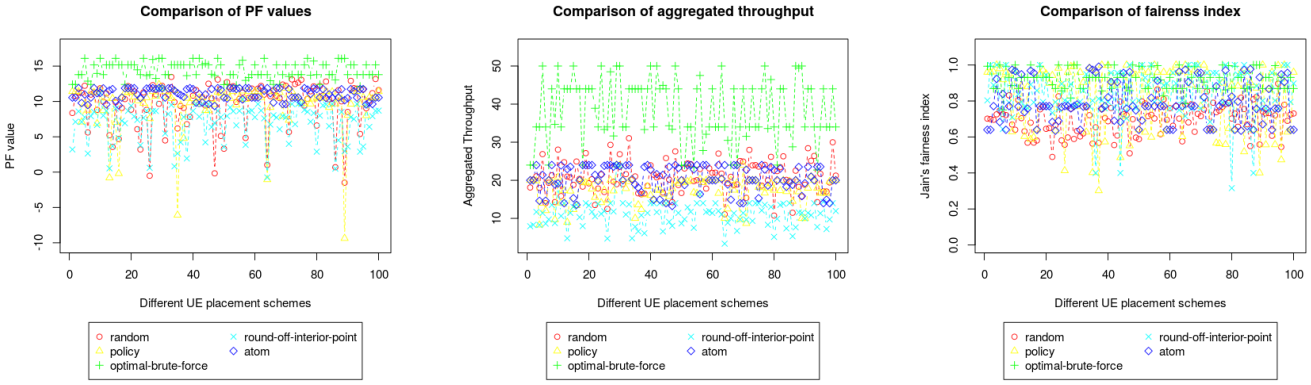
(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 6. The distance of various previous schemes to the optimal (5 UEs and 3 APs, P_s = 0.8, pf only).



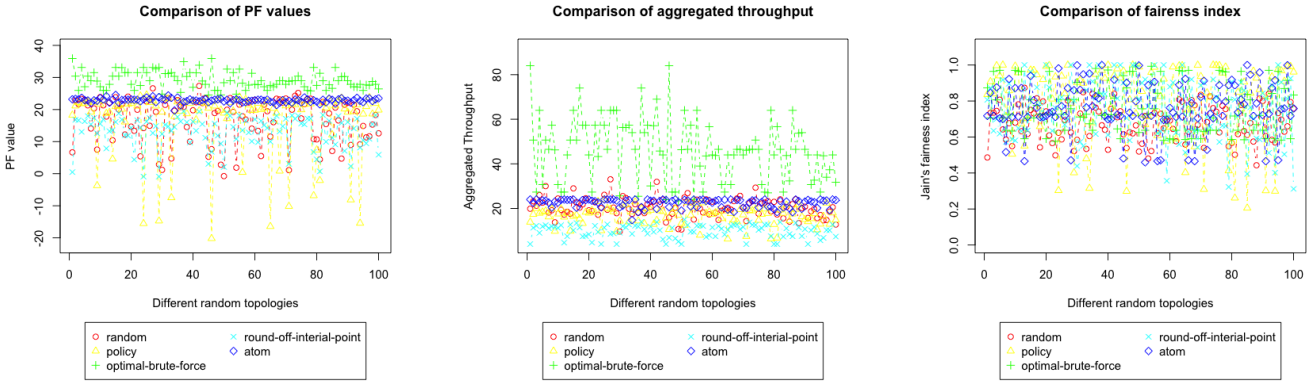
(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 7. The distance of various previous schemes to the optimal (10 UEs and 3 APs, P_s = 0.8, pf only).



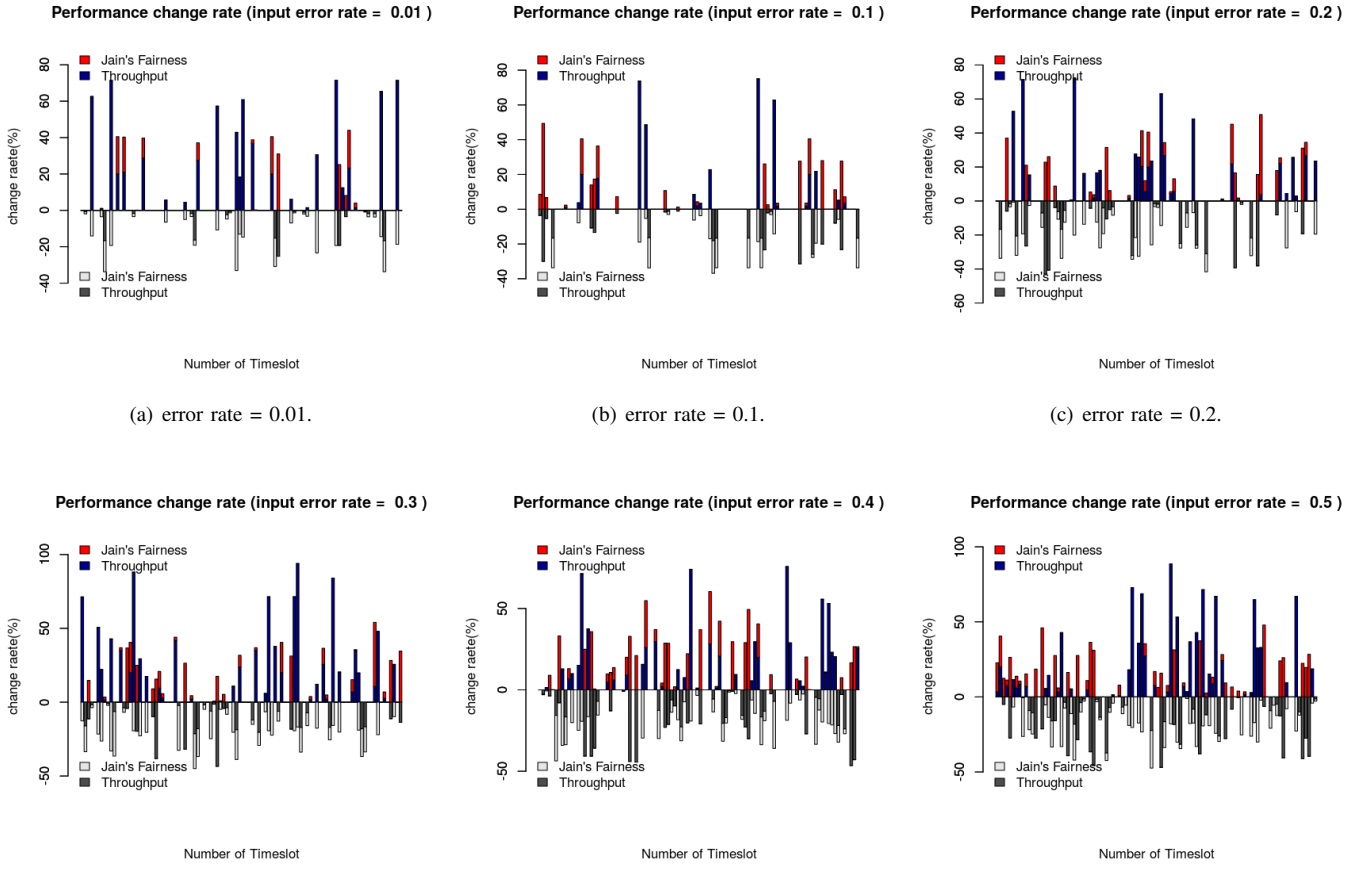
(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 8. The distance of various previous schemes to the optimal (5 UEs and 3 APs, P_s = 0.8, both).



(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

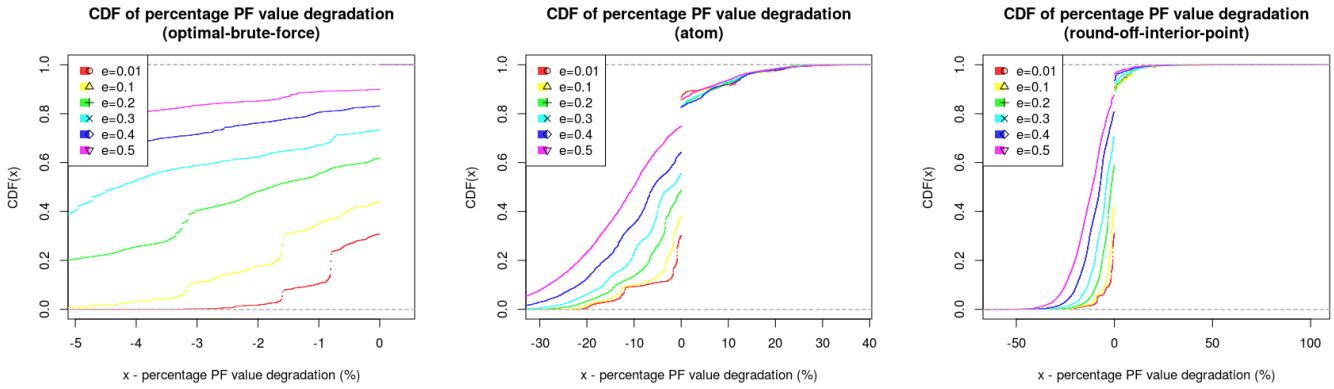
Fig. 9. The distance of various previous schemes to the optimal (10 UEs and 3 APs, P_s = 0.8, both).



(a) error rate = 0.01. (b) error rate = 0.1. (c) error rate = 0.2.

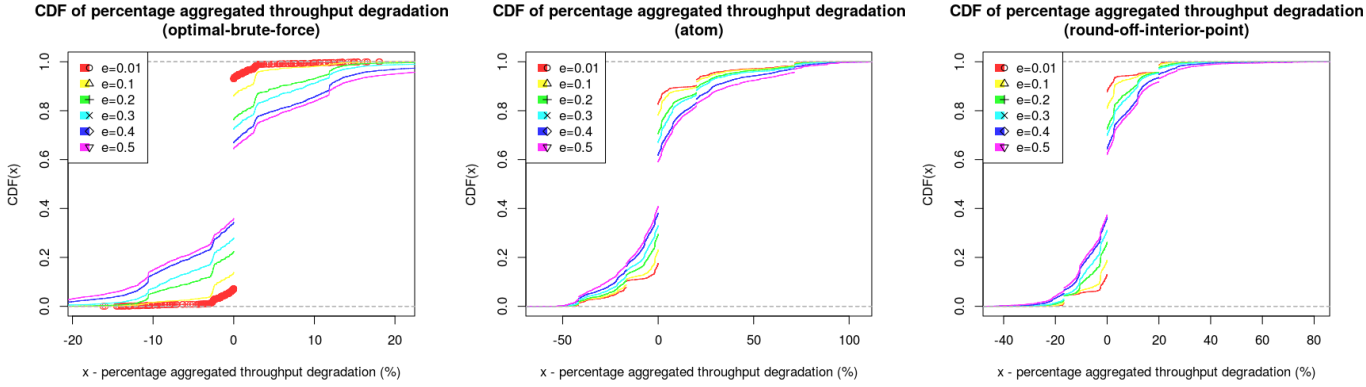
(d) error rate = 0.3. (e) error rate = 0.4. (f) error rate = 0.5.

Fig. 17. Overlayed performance change for 100 sampled time slots. (5 UEs and 3 APs, P_s = 0.8, pf only).



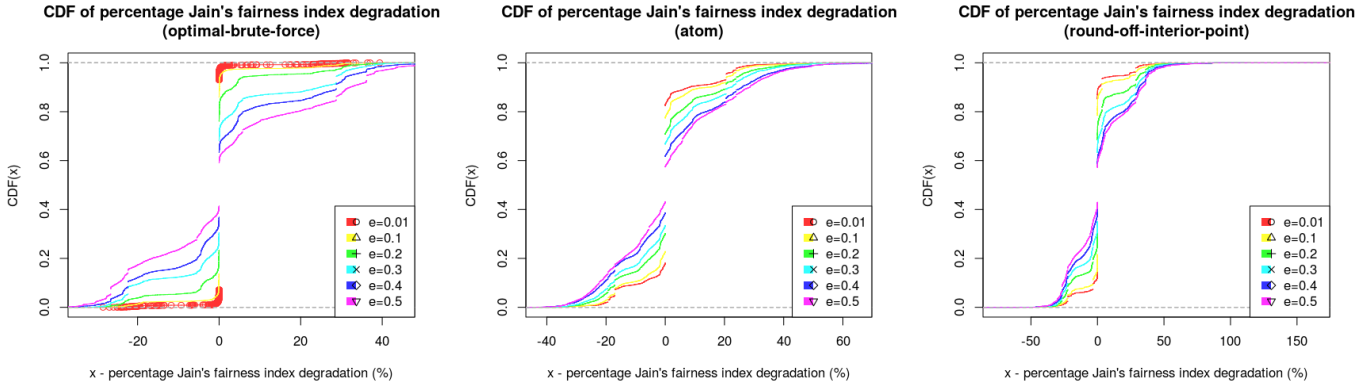
(a) optimal-brute-force. (b) atom. (c) round-off-interior-point.

Fig. 18. Impact of throughput estimation error to PF value (5 UEs and 3 APs, P_s = 0.8, pf only).



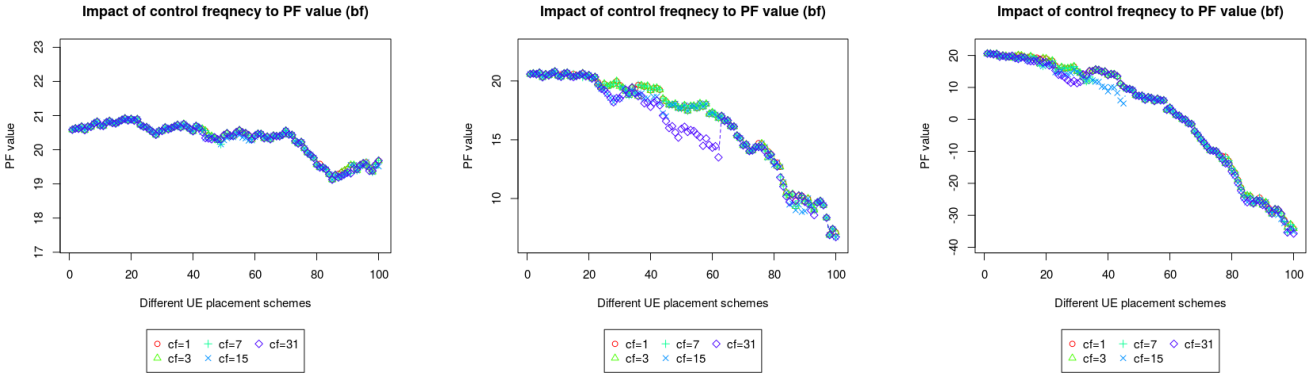
(a) optimal-brute-force. (b) atom. (c) round-off-interior-point.

Fig. 19. Impact of throughput estimation error to aggregated throughput (5 UEs and 3 APs, P_s = 0.8, pf only).



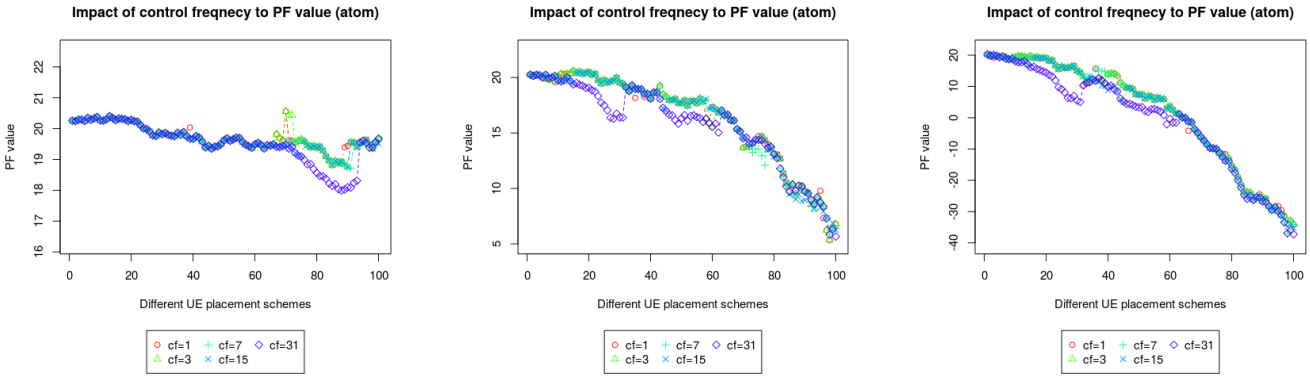
(a) optimal-brute-force. (b) atom. (c) round-off-interior-point.

Fig. 20. Impact of throughput estimation error to Jain's fairness metric (5 UEs and 3 APs, P_s = 0.8, pf only).



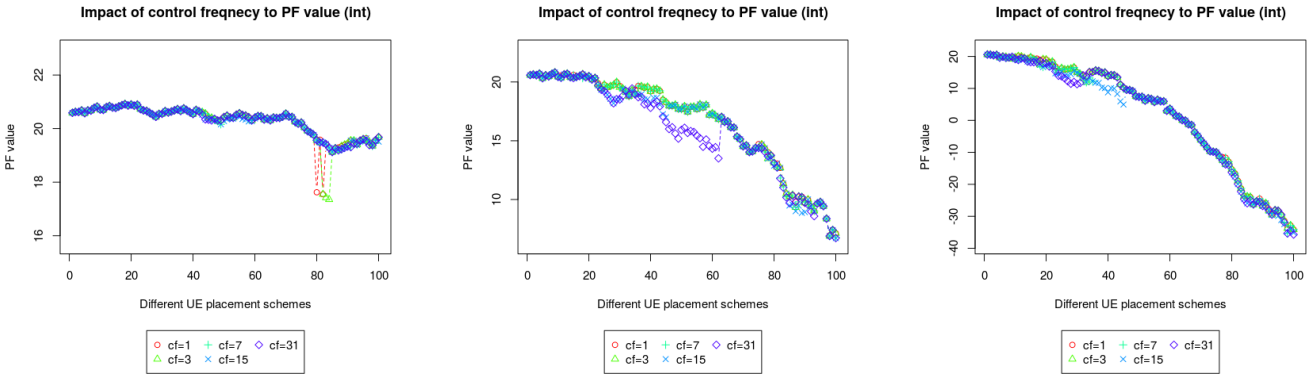
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 21. Impact of control frequency to the PF value of optimal-brute-force (5 UEs and 3 APs, P_s = 0.8, pf only).



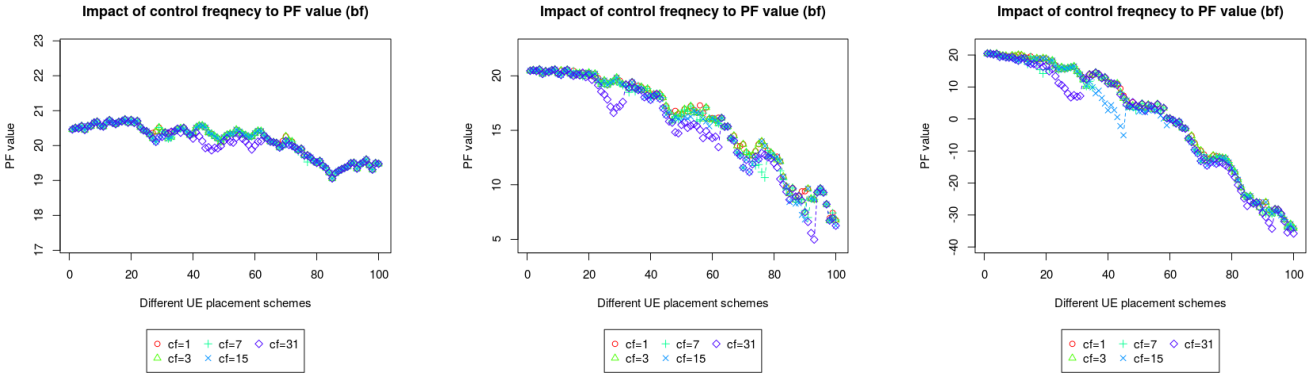
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 22. Impact of control frequency to the PF value of ATOM (5 UEs and 3 APs, P_s = 0.8, pf only).



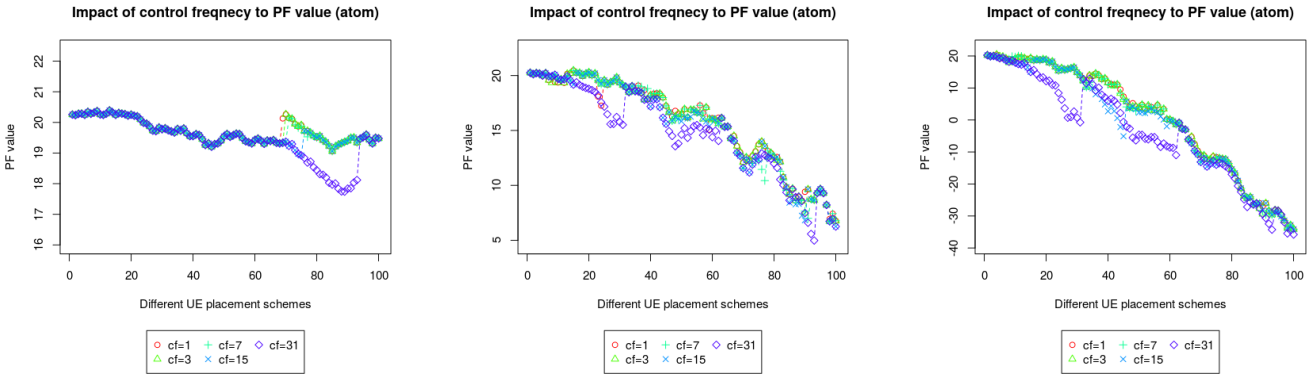
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 23. Impact of control frequency to the PF value of round-off-interior-point (5 UEs and 3 APs, P_s = 0.8, pf only).



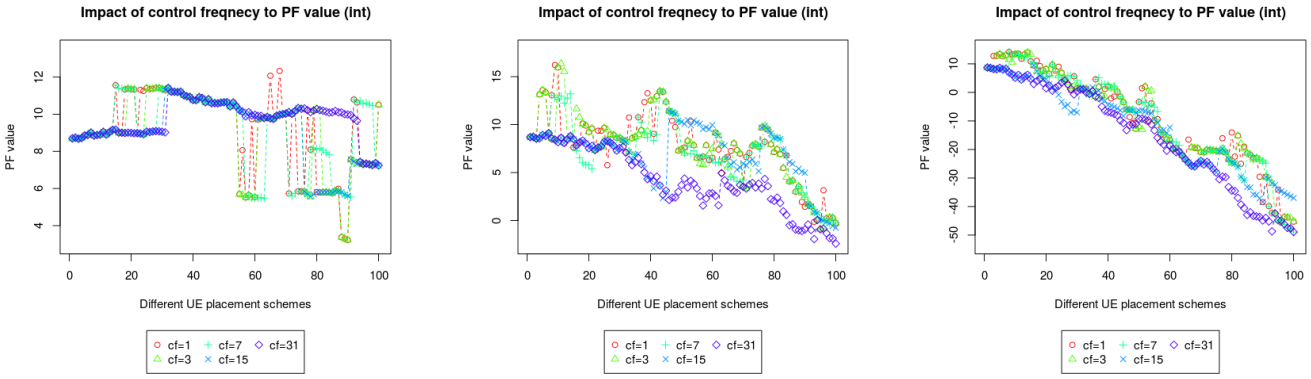
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 24. Impact of control frequency to the PF value of optimal-brute-force (5 UEs and 3 APs, P_s = 0.8, both).



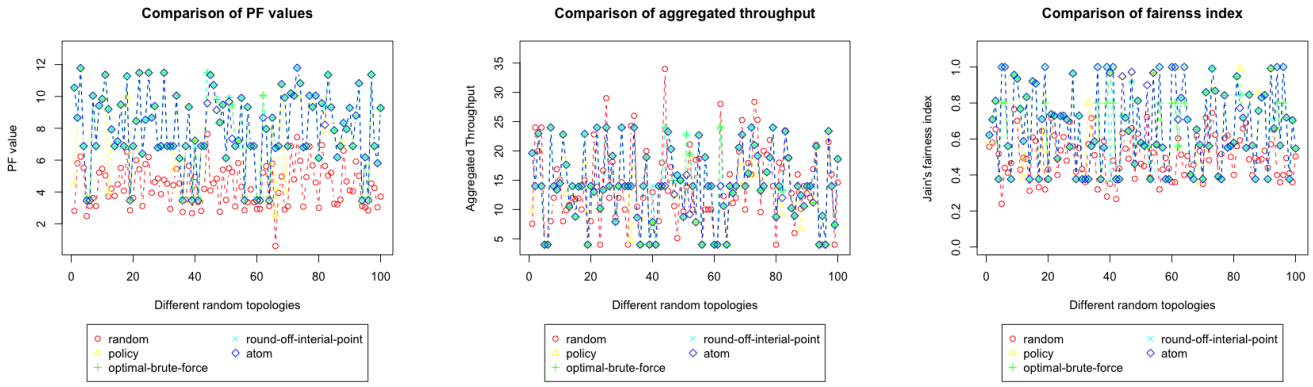
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 25. Impact of control frequency to the PF value of ATOM (5 UEs and 3 APs, P_s = 0.8, both).



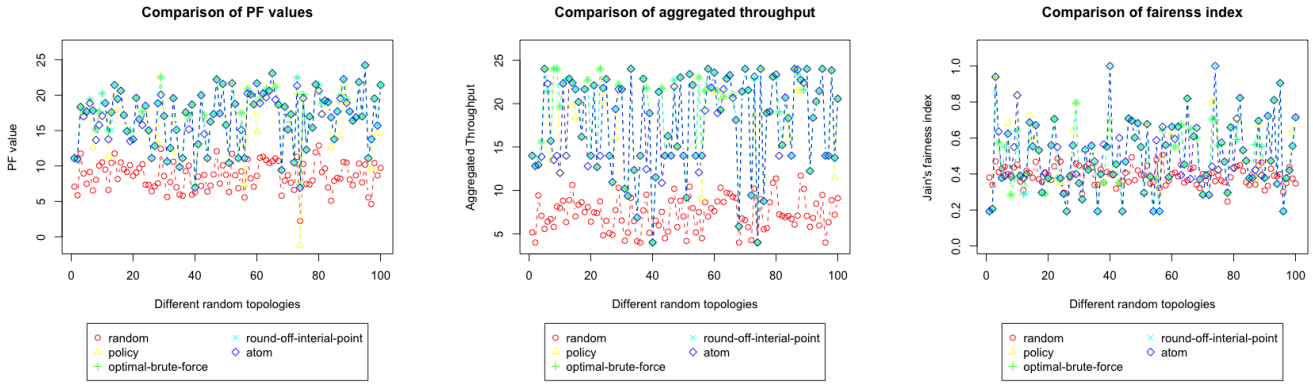
(a) c = 10% (b) c = 30% (c) c = 50%

Fig. 26. Impact of control frequency to the PF value of round-off-interior-point (5 UEs and 3 APs, P_s = 0.8, both).



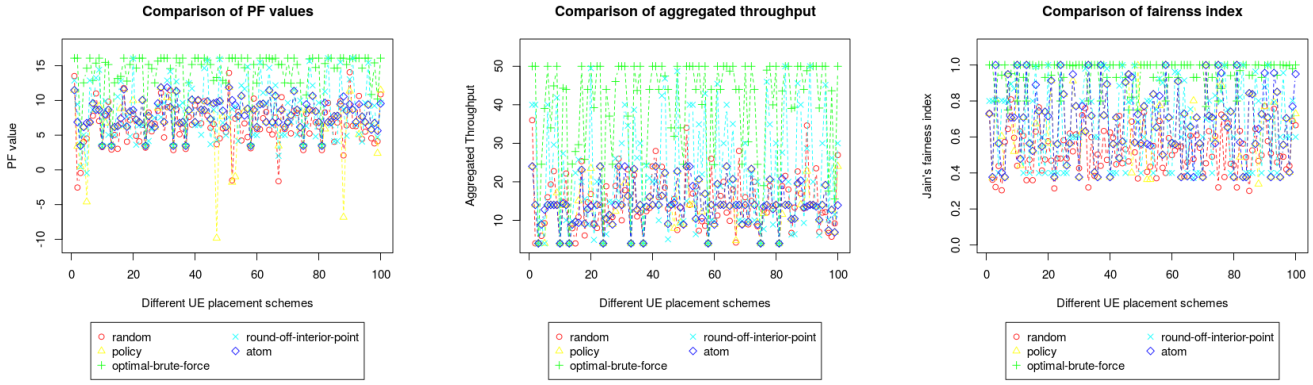
(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 27. The distance of various previous schemes to the optimal (5 UEs and 3 APs, P_s = 0.2, pf only).



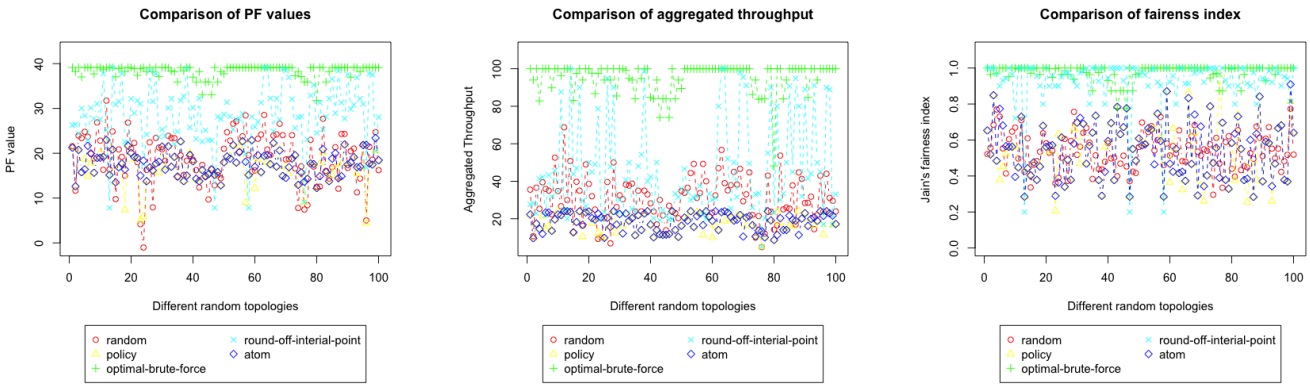
(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 28. The distance of various previous schemes to the optimal (10 UEs and 3 APs, P_s = 0.2, pf only).



(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 29. The distance of various previous schemes to the optimal (5 UEs and 3 APs, P_s = 0.2, both).



(a) PF value. (b) Aggregated throughput. (c) Jain's fairness.

Fig. 30. The distance of various previous schemes to the optimal (10 UEs and 3 APs, P_s = 0.2, both).