Stable interaction of self-optimization processes in wireless networks

Citation for published version (APA):

DOI:
10.1109/iccw.2011.5963592

Document status and date:
Published: 01/01/2011

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
https://www.tue.nl/index.php?id=71870

Take down policy
If you believe that this document breaches copyright please contact us at: openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 12. Feb. 2019
Stable interaction of self-optimization processes in wireless networks

Markus Gruber*, Sem Borst†, and Edgar Kuehn‡

Abstract—The ability of base stations in wireless access networks to regularly and autonomously self-optimize their parameters has become a key requirement from network operators. A number of specific optimization use cases have been discussed whose interaction conflicts can be mitigated by a separation in individual groups that are optimized consecutively with negligible or no mutual interactions within a group. In this paper, we introduce coordination and separation strategies and discuss their suitability to avoid interaction conflicts. In particular, we discuss a separation by the amount of measurements needed to make a reasonable decision to modify the parameters. For this case, we derive insight into the statistical relation between the tolerances we accept and the number of observations that are necessary to trigger a parameter modification. Recommendations are provided towards a stable holistic autonomous solution for wireless access networks.

Index Terms—self-optimizing networks, self-organizing networks, holistic optimization, interaction conflicts, LTE

I. INTRODUCTION

In wireless access networks, base stations typically provide a large number of parameters that can be set and modified by the network operator [1]. This kind of flexibility is needed to accommodate base stations to different environments and conditions. On the other hand, manually setting and adapting these parameters is not a cost-effective option for the operator so that substantial research effort has been spent to provide autonomous solutions. For the LTE (Long Term Evolution) network standardized by the 3rd Generation Partnership Project (3GPP) [2], this challenge is approached from a use-case perspective [3]. Examples for such use cases are coverage and capacity optimization, energy savings or mobility robustness optimization, to name just a few. The ultimate goal is to have a completely autonomous network with very little or preferably no need for human intervention (“zero touch”).

A key issue that naturally arises in such a use case centric approach is that the various use cases are optimized on an individual basis, without taking into account potential effects on other use cases. In other words, the presented use cases are not independent of each other and the modification of a parameter beneficial for one use case may have a negative impact on another use case. We will call this an interaction conflict. Conflicts that may appear in parallel and distributed environments are discussed at a very general level in [4]. Joint optimization for specific problems in wireless access networks has been treated in [5-6], whereas LTE-related conflicts are investigated in [7-8].

The Socrates project consortium funded by the European Union proposes to avoid interaction conflicts by a scheme where use cases trigger each other in a consecutive order [9]. In situations where this is not feasible, multiple use cases cooperate/co-act in a use case bundle which is recommended to be treated as one optimization problem. Moreover, they proposed to categorize interaction problems into two classes, namely parameter value conflicts and metric value conflicts.

- Parameter value conflicts occur if any two use cases have access to the same control parameter.
- Metric value conflicts occur if any two use cases influence a common metric that is used as feedback information to influence either use case.

This paper focuses on coordination and separation strategies for optimization processes. We define a process to be an algorithm that optimizes the control parameters of a given use case. Section II discusses coordination strategies whereas Section III focuses on strategies how optimization processes can be reasonably separated in different groups. Section IV then analyzes a specific separation strategy from a statistical perspective and Section V discusses the main conclusions and recommendations of our work.

II. COORDINATION STRATEGIES

In this section we analyze possible types of coordination strategies that interacting optimization processes may adopt. In this context we distinguish between coordination by information where optimization processes are on the same hierarchical level and coordination by control where one optimization process imposes rules on a second optimization process. A further aspect that is applicable to both these cases is the fact that coordination can be either unidirectional or typically yielding more complexity - bidirectional.

A. Coordination by information

In this case optimization process A provides information to optimization process B where process B is able to utilize this information for its own benefit. An example would be the interaction between the random access channel (RACH) optimization use case and the mobility robustness optimization
(MRO) use case specified in [3].

The random access channel provides resources for terminals that go from idle to connected state with respect to a given base station. This channel is also relevant for mobility since terminals entering a new cell need to get connected before the user communication can continue. The RACH optimization use case addresses the question how many resources should be allocated to this channel. New calls and handovers may be blocked if there are too few resources allocated, whereas the overall spectral efficiency will be low if too many resources are already reserved.

The MRO use case involves the optimization of parameters such that both the handover failure rate is minimized and that situations where terminals are handed over and back in a short time (“ping-pong”) are avoided. For the latter goal, a so-called time-to-trigger value imposes a resting time during which the handover criteria have to be continuously fulfilled before the actual handover is triggered. If this time is too short, the probability of ping-pong effects is high, whereas for too long time-to-trigger values the terminal may have already moved to very bad reception conditions and the handover failure rate increases.

However, the interaction between RACH optimization and MRO can be critical: Let us assume that at a given time the RACH is at its capacity limit and unable to accommodate any new terminals. Let us further assume that the RACH optimization process does not react to this situation for some reason (e.g. because this situation is expected to be relatively short or because it is difficult to allocate further resources). Then the MRO use case will detect a higher handover failure rate and potentially decrease the time-to-trigger value. This in turn would have the effect that there are even more handovers, some of them due to ping-pong effects, and that there would be even more access attempts on the RACH – an instability with a self-reinforcing feedback loop.

This situation can be avoided if the RACH optimization process informs the MRO process about the RACH being fully loaded. With this information, the MRO process is able to differentiate the typical case where a higher handover failure rate should lead to a smaller time-to-trigger value and the pathological case where a higher handover failure rate is due to a RACH overload and the time-to-trigger must not be decreased. Please note that this use case coordination may be applicable to a single base station, but may as well span across several base stations, in which case it can be helpful that dedicated RACH resources may be communicated between base stations during the handover procedure.

B. Coordination by control

For the case of coordination by control, let us consider MRO and the specific mobility load balancing optimization (LB) use case where load is balanced within the confines of the same technology and the same frequency band by shifting the handover region and thus forcing terminals from one cell to the other. Technically, this is achieved by changing the SINR (signal to interference plus noise ratio) requirements for the handover such that terminals are transferred in a way that does not conform to the optimal handover procedure. Correspondingly, if there are no coordination mechanisms between MRO and LB, MRO could try to move the handover region back to what is optimal in terms of MRO. LB, in turn, will then try to compensate this again and so forth, i.e. the result is system instability due to oscillation.

This situation can be solved by giving LB a higher priority than MRO. Thus, LB controls MRO by overruling parameter changes, by permitting only a limited range of parameters, or even by completely blocking changes of specific parameters. By the subordination of MRO to LB, a conflict situation resulting in oscillation is avoided.

III. Separation Strategies

After the discussion of coordination strategies, we will now focus on the question how optimization processes can be reasonably separated to avoid conflicts. We define optimization processes to be separate if their interaction can be neglected because they work on different domains. One such domain can be the timescale of re-optimization. For instance, an optimization process that re-optimizes its parameters once a month will not be influenced by an optimization process that re-optimizes once an hour. On the other hand, the hourly re-optimization takes the monthly re-optimization as an outside constraint that cannot be changed. Thus, the separation ensures that oscillations and instabilities do not occur. Secondly, optimization processes that have the same timescale of re-optimization should be re-optimized at the same time and not in a staggered way; this ensures that interaction problems are only observed and treated at pre-defined points in time. Use cases of wireless access networks can vary widely in their timescale of re-optimization and can thus be easily separated accordingly. However, the question arises how these timescales are best determined. For the timescales of re-optimization, there are several strategies that we will discuss in the following.

A. Cost of parameter modification

One constraint that needs to be considered when determining the timescale of re-optimization is the cost of a parameter modification. The most costly scenario for a network operator is a situation where parts of the system need to be mechanically moved. Typically, parts can tolerate a certain number of movements before they fail, then they have to be replaced involving manual intervention; a second aspect is the failure time itself including service degradations or outage which is critical for the reputation of network operators in a competitive environment. An example for a case involving moving parts may be the direction of an antenna and obviously operators would only be willing to re-optimize when really necessary.

A class of processes where parameter modifications come along with moderate costs are those where parameter changes are communicated to terminals on an individual basis by RRC (Radio Resource Control) signaling. This is for example the case for the mobility parameters where additional radio resources have to be provided for the transmission of the new
parameters.

Very little cost is expected where parameter changes are communicated to terminals via broadcast messages. These messages are distributed in any case, with or without parameter changes, so no additional radio resources have to be taken into account. A corresponding example would be the assignment of a carrier frequency that is broadcast in a system information block to all terminals alike.

B. Number of measurements needed

Another way to find a reasonable timescale of re-optimization is to consider how many measurements/observations are needed to take a statistically meaningful decision. Let us for example assume that the relevant metric for a given optimization process is poor within a given measurement interval. The question now is whether the poor performance is simply due to noise or whether there is actually a reason to re-optimize. Since different optimization processes rely on different kinds of measurements, the number of needed observations is a natural way to separate optimization processes in different timescales. In this context, the end of a measurement interval triggers the decision for re-optimization.

IV. MEASUREMENTS AND ADMISSIBLE METRIC RANGES

In this section we will analyze how many measurements/observations, i.e., samples, are needed to come to a statistically meaningful decision whether to re-optimize or not. This will naturally depend on the metric range we admit without triggering a change. In this context we will look at two scenarios, one without fixed measurement intervals (relevant for optimization processes that control/trigger other processes as discussed in Section IIB) and one with fixed measurement intervals (relevant for the separation of optimization processes in time as discussed in Section IIII). In both scenarios we do not continuously optimize, but we only do so if the current measurements reflect that the relevant metric has dropped significantly. This takes into account that a certain cost is associated with each re-optimization as discussed in Section IIIA and that we do not want to spend any effort unless the improvement in the metric is significant.

In the context of the following analysis, let \( m(\zeta; \theta) \) be some performance metric of the system as function of some parameter \( \zeta \) characterizing the ambient system conditions and some set of control parameters \( \theta \). In particular, we could have \( \theta = (\theta_1, \ldots, \theta_k) \), with each of the \( \theta_k \) representing a particular subset of the control parameters based on some logical or functional categorization. We aim to determine the control parameters \( \theta \) so as to optimize the performance metric \( m(\zeta; \theta) \), e.g., \( \theta^* = \arg \max_{\theta \in \Theta} m(\zeta; \theta) \), with \( \Theta \) denoting the set of admissible settings for \( \theta \).

In practice, the functional relationship \( m(\cdot) \) is typically not known in explicit form with any degree of accuracy, and can only be learned through measurements/observations. Also, the value of \( \zeta \) will usually vary over time and not be known exactly. Yet, similar to stochastic approximation, we assume that a procedure is available for determining the optimal control parameters \( \theta^*(\zeta) \) for a given value of \( \zeta \) and obtaining the corresponding optimal value of the performance metric \( m^*(\zeta) = m(\zeta; \theta^*(\zeta)) = \max_{\theta \in \Theta} m(\zeta; \theta) \). However, we wish to invoke this procedure judiciously and only trigger a re-optimization of \( \theta \) when the current value of \( \zeta \) has reasonably changed, causing \( m(\zeta_{current}; \theta^*(\zeta_{earlier})) \) to significantly drop below \( m(\zeta_{earlier}) = m(\zeta_{earlier}; \theta^*(\zeta_{earlier})) \).

A. Flexible measurement intervals for coordination

In this section we will focus on optimization processes that can spontaneously re-optimize whenever suitable. The advantage of this class of processes is that the re-optimization is triggered when it is most appropriate and not after given intervals. However, this approach can only be applied if optimization processes are not separated in time, but rather collaborate by triggering each other. We will sketch a solution based on a Hidden Markov Model (HMM) in which we abstract the system to states that are relevant for re-optimization. For this purpose we will need to have some knowledge about the probability that a change in system conditions from one observation to the next one should cause a modification of parameters (in order to determine the transmission probabilities of the HMM). This probability can for instance be derived from experience from real systems.

Let us assume a HMM with the hidden states “re-optimization not necessary” and “re-optimization beneficial” (Fig. 1). In reality we do not know which of these two hidden states the system can be mapped to. Instead, we can only check whether the currently observed metric \( m(\zeta_{current}; \theta^*(\zeta_{earlier})) \) is above or below a dynamic threshold value \( m(\zeta_{earlier}; \theta^*(\zeta_{earlier})) - \gamma \) which is continuously updated with respect to the last achieved metric value. In other words, we know whether the system is in state “above (or equal) metric threshold” or state “below metric threshold” and we can base our decisions whether to re-optimize or not only on these observations, but we cannot know whether our decision actually makes sense or whether it is only caused by random fluctuations of the metric.

The probabilities that the observations are caused by the hidden states “re-optimization not necessary” and “re-optimization beneficial”, respectively, can be described by the emission probabilities \( p_1 \) and \( p_2 \) (\( p_1 \) is the probability of the metric to be below the threshold although a re-optimization is not necessary and \( p_2 \) is the probability of the metric to be above (or equal) the threshold although a re-optimization would be beneficial). The transition probabilities \( a_1 \) and \( a_2 \) must have been learned from the system before, i.e. the average timescale for which a re-optimization is not necessary must be known. Finally, we can apply the Viterbi algorithm to determine the most probable state path and decide whether a re-optimization should be carried out or not. This decision can be made for each individual observation so that the reaction time of the system will be quick.
B. Fixed measurement intervals for separation

In this section we will focus on optimization processes that can only be re-invoked after certain given time intervals, e.g. to enforce a clear separation in time from other processes. The time interval must incorporate enough measurements to make a statistically meaningful decision whether to re-optimize or not. As opposed to the HMM approach, no a priori knowledge of a reasonable frequency of re-optimizations is needed.

Let us consider a measurement interval with \( N \) samples of the performance metric of interest, and let the random variable \( X_i \) denote the value observed at the \( i \)-th sample. Assuming the parameters \( \zeta \) and \( \theta \) to be fixed during the measurement interval, the empirical average \( Y = \frac{1}{N} \sum_{i=1}^{N} X_i \) provides an (unbiased) estimate of \( m(\zeta; \theta) \), i.e., \( E[Y] = m(\zeta; \theta) \). For compactness, denote by \( m^* = m^*(\zeta_{\text{earlier}}) \) the optimal value of the performance metric as determined at the most recent execution of the optimization procedure. We then assume that the value of \( \theta \) gets modified if \( Y < m^* - \gamma \) and remains unchanged otherwise, with \( \gamma \) a statistical error margin.

The question that arises is how to set the values of \( \gamma \) and \( N \) in order to achieve two key requirements:

1) avoid (unnecessary) re-optimizations: the probability of a “false positive” (type-I error which equals the emission probability \( p_1 \) in the HMM) should be small. Specifically, if the current value of \( \theta \) is correct, then the probability of modifying \( \theta \) due to random fluctuations should be below some threshold \( \epsilon_1 \), i.e.,

\[
p_1 = P[Y < m^* - \gamma | \zeta; m(\zeta; \theta) = m^*] \leq \epsilon_1
\]

2) perform re-optimizations when necessary: the probability of a “false negative” (type-II error which equals the emission probability \( p_2 \) in the HMM) should be small. Specifically, if the current value of \( \theta \) is incorrect, then the probability of \( \theta \) remaining unchanged should be below some small threshold \( \epsilon_2 \), i.e.,

\[
p_2 = P[Y \geq m^* - \gamma | \zeta; m(\zeta; \theta) \leq m^* - \delta] \leq \epsilon_2
\]

To determine the above probabilities, we need to know the distribution of the empirical average \( Y \). Assuming the parameters \( \zeta \) and \( \theta \) to be fixed during the measurement interval, it is reasonable to assume that the random variables \( X_i \) are approximately independent, provided the sampling epochs are sufficiently separated in space/time relative to the spatial/temporal correlation in the performance metric of interest. For large values of \( N \), the central limit theorem then implies that

\[
\frac{\sum_{i=1}^{N} X_i - N m}{\sigma \sqrt{N}} \text{ has an approximately standard normal distribution, with } m = E(X_i) \text{ and } \sigma = \sqrt{E[(X_i - m)^2]}
\]

denoting the mean and standard deviation of \( X_i \), respectively. Denote by \( Z \) a standard normal random variable. It follows that

\[
p_1 = P\left[ Z < \frac{-\gamma \sqrt{N}}{\sigma_1} \right] = \Phi\left( \frac{-\gamma \sqrt{N}}{\sigma_1} \right)
\]

and

\[
p_2 = P\left[ Z > \frac{(\gamma - \delta) \sqrt{N}}{\sigma_2} \right] = 1 - \Phi\left( \frac{(\gamma - \delta) \sqrt{N}}{\sigma_2} \right)
\]

with

\[
\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-u^2/2} du
\]

denoting the cumulative distribution function of the standard normal random variable \( Z \).

In order to (approximately) satisfy the requirement \( p_1 \leq \epsilon_1 \), we need to set the parameters \( \gamma \) and \( N \) such that

\[
\Phi\left( \frac{-\gamma \sqrt{N}}{\sigma_1} \right) \leq \epsilon_1, \text{ i.e.,} \gamma \sqrt{N} \geq \sigma_1 \Phi^{-1}(1 - \epsilon_1)
\]

Thus

\[
\gamma \geq \frac{\sigma_1 \Phi^{-1}(1 - \epsilon_1)}{\sqrt{N}} \text{ or equivalently } N \geq \frac{\sigma_1^2 \Phi^{-1}(1 - \epsilon_1)^2}{\epsilon_1^2}
\]

(1)

In order to (approximately) meet the requirement \( p_2 \leq \epsilon_2 \) we further need to set \( \gamma < \delta \), and

Fig. 1: Abstraction of the system to states that are relevant for re-optimization: Hidden Markov Model with hidden states „Re-optimization not necessary” and „Re-optimization beneficial”. The actual decisions are based on the observations “Above (or equal) metric threshold” and “Below metric threshold”. \( a_1 \) and \( a_2 \) are the transition probabilities that need to be known in advance (e.g. by learning/experience).
1 - \Phi \left( \frac{(\gamma - \delta)\sqrt{N}}{\sigma_2} \right) \leq \epsilon_2, \text{ i.e.,} \\
(\delta - \gamma)\sqrt{N} \geq \sigma_2 \Phi^{-1}(1 - \epsilon_2).

Thus
\delta - \gamma \geq \frac{\sigma_2 \Phi^{-1}(1 - \epsilon_2)}{\sqrt{N}} \text{ or equivalently}
N \geq \frac{\sigma_2^2 \Phi^{-1}(1 - \epsilon_2)^2}{(\delta - \gamma)^2}. \quad (2)

Note that the required sample size $N$ attains a minimum when
\[ \frac{\sigma_1 \Phi^{-1}(1 - \epsilon_1)}{\gamma} = \frac{\sigma_2 \Phi^{-1}(1 - \epsilon_2)}{\delta}, \]
and equals
\[ N_{\text{min}} = \frac{\left( \sigma_1 \Phi^{-1}(1 - \epsilon_1) + \sigma_2 \Phi^{-1}(1 - \epsilon_2) \right)^2}{\delta^2}. \quad (3) \]

The above two conditions give rise to a trade-off between accuracy (smaller $\delta$ and hence correspondingly smaller $\gamma$), which is important in a more static situation, and responsiveness (smaller $N$), which is crucial in a more dynamic environment, where the ambient system conditions as captured by the parameter $\zeta$ may unpredictably vary over time.

Now observe that the standard deviations $\sigma_1$ and $\sigma_2$ are unknown, which makes it difficult to determine the required number of observations $N$ for given values of $\delta$ and $\gamma$. However, if the number of observations $N$ is fixed, then we can determine a suitable value for $\gamma$ by using the empirical standard deviation as an estimate for $\sigma$:
\[ \hat{\gamma} = \frac{\sigma \Phi^{-1}(1 - \epsilon_1)}{\sqrt{N}} \text{ with} \]
\[ \hat{\sigma} = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2. \]

We now present some numerical results. For convenience, we assume that the standard deviation of $X_i$ is independent of the parameters $\zeta$ and $\theta$, so that $\sigma_1 = \sigma_2 = \sigma$. Now observe that the various requirements for the sample size $N$ and the parameters $\delta$ and $\gamma$ only depend on the ratios $\hat{\delta} = \frac{\delta}{\sigma}$ and $\hat{\gamma} = \frac{\gamma}{\sigma}$. Hence we will consider the required sample size $N$ as function of these scaled quantities.

In the first two experiments we set the thresholds for the “false positive” and “false negative” probabilities to $\epsilon_1 = 0.1$ and $\epsilon_2 = 0.1$. Fig. 2 plots the required sample size $N$ as function of the relative statistical error margin $\hat{\gamma}$ for a fixed relative tolerance $\delta = 0.1$. The left branch in the figure corresponds to the probability of false positives as captured in (1), while the right branch pertains to the probability of false negatives as reflected (2). Note that the required sample size substantially grows when the statistical error margin tends to 0 or approaches the tolerance level. Indeed, the required sample size grows as $O\left( \frac{1}{\hat{\gamma}^2} \right)$ as $\hat{\gamma} \downarrow 0$ in the left branch as governed by (1) and behaves as $O\left( \frac{1}{(\hat{\delta} - \hat{\gamma})^2} \right)$ as $\hat{\gamma} \uparrow \hat{\delta}$ in the right branch as described by (2). This may be explained from the fact that the likelihood of false positives and false negatives rises sharply in these two cases, and hence a large sample size is needed to keep these probabilities below the threshold values. Fig. 3 shows the minimum required value of the sample size $N$ as function of the relative tolerance level $\hat{\delta}$. In the next two experiments (Fig. 4 and Fig. 5) we reduce the thresholds for the “false positive” and “false negative” probabilities to $\epsilon_1 = 0.01$ and $\epsilon_2 = 0.01$.

Fig. 2: Required sample size $N$ as function of relative statistical error margin $\hat{\gamma}$; $\hat{\delta} = 0.1$, $\epsilon_1 = 0.1$, $\epsilon_2 = 0.1$.

Fig. 3: Minimum required value of sample size $N$ as function of relative tolerance level $\hat{\delta}$; $\epsilon_1 = 0.1$, $\epsilon_2 = 0.1$. 
in wireless access networks from a statistical perspective. Although the tools employed to this end are standard, this analysis has concrete implications for the design of optimization processes in wireless access networks. In particular, re-optimizations should not be limited to abnormal cases where key performance indicators are entirely unacceptable, but should rather be carried out whenever necessary from a performance point of view.

ACKNOWLEDGMENT

Part of the research leading to these results has been performed within the UniverSelf project (www.UniverSelf-project.eu) and received funding from the European Community’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 257513. We would like to explicitly thank Iraj Saniee and Dietrich Zeller, both from Alcatel-Lucent Bell Labs, for many fruitful discussions.

REFERENCES