Optimal Usage of Multiple Network Connections

Anne Setämaa-Kärkkäinen
University of Jyväskylä
P.O. Box 35 (Agora)
FI-40014 University of Jyväskylä, Finland
annseta@mit.jyu.fi

Jani Kurhinen
University of Jyväskylä
P.O. Box 35 (Agora)
FI-40014 University of Jyväskylä, Finland
kurhinen@mit.jyu.fi

ABSTRACT

In the future mobile networks, a mobile terminal is able to select the best suitable network for each data transmission. The selection of a network connection to be used has been under a lot of study. In this paper, we consider a more extensive case in which we do not select a network connection but use several network connections simultaneously to transfer data. When data is transferred using multiple network connections, a network connection has to be selected for each component of the data. We have modelled this problem as a multiobjective optimization problem and developed a heuristic to solve the problem fast in a static network environment. In this paper, we discuss solving the problem in a dynamic network environment in which the availability as well as the rate and prizing of connections vary. We introduce an improved version of the heuristic that reacts to changing network conditions and improves the solution when possible.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; G.1.6 [**Optimization**]: Integer programming

General Terms

Algorithms

Keywords

Wireless networks, network connection selection, scheduling, multiobjective optimization, heuristics

1. INTRODUCTION

The number of mobile users and mobile services available is increasing fast. The increasing amount of transferred data has created a need for new faster types of transmission methods. However, the faster the transmission rate is

the smaller the wireless transmission range usually is. This progress forces terminal manufacturers to design multimode devices that can operate using several different kinds of network technologies. When these versatile terminals are used within an environment where there exist several network access providers, the users of mobile terminals can choose both the network access technology and the network access provider.

In this paper we consider a situation in which it is possible to use several different kinds of networks from different access providers simultaneously. In order to enable using several connections at the same time, an intelligent connection manager is needed to control the multiple seamless connections. This issue has been studied earlier mainly from the network-level point of view. However, there is also a need to take into account user-level parameters when making a decision on the network connections used. We have considered the network connection selection from the users point of view. The user usually wants to transfer data as fast and as cheaply as possible. Therefore, we consider the rates and the prices of the connections as the selection criteria. Also other issues could be taken into account since the solution method developed is general.

The selection of network connections is made using optimization. The optimization problem considered is which connection to use for transferring each component of the data. We call this problem the *network connection selection problem* (NCS) and it is formulated as a multiobjective optimization problem where the objective is to minimize both the costs and the time used in transferring. The NCS problem was first introduced in [10], in which the problem was formulated as a multiobjective optimization problem and different multiobjective optimization methods were compared for solving the problem. In [11] a heuristic for solving the problem in a static environment was presented. In this paper, we present an improved dynamic version of the heuristic that reacts to changing network conditions. We also discuss some technical issues involving the NCS problem.

This paper is organized as follows. In the next section, we discuss technical issues of using multiple network connections simultaneously. In Section 3, the network connection selection problem is modelled as a multiobjective optimization problem. Solving the problem is discussed in Section 4 where a heuristic algorithm developed for solving the problem in a static environment is briefly described. Dynamic environment is considered in Section 5 and the heuristic is modified to deal with changing network conditions. Sec-

tion 6 presents an illustrative example, and conclusions are drawn in Section 7.

2. NETWORK CONNECTION SELECTION

A current trend in networking is the convergence of different networks, including the process of wired and wireless networks joining together and forming a single fourth generation (4G) network. The 4G networks are not dependent on a certain frequency or a modulation method as the current generation of networks is [2]. The 4G networks are based on the idea of roaming between networks, even though those networks do not share the same technology. The mobility between different technologies is called vertical roaming, and it is needed when designing multi-technology networks. One downside of this mobility between different types of networks is that an access terminal will become more complicated, since it has to be able to receive and transmit data using different frequencies and modulations.

A technology that could provide an answer to this problem is called software defined radio [9]. With the software defined radio, the radio transceiver module is not hardwired to decode one particular transmission technology, but it has the ability to change the communication parameters. In this kind of a system, it is possible to define separate realizations of different types of transceivers as software which can be saved in the memory of the device. When a different type of network access is needed, one (or many, depending on the situation) of the stored transceivers can be loaded to be actively used. With programmable hardware this active transmitter/receiver system reconfigures itself at a hardware level to be able to execute efficiently the selected communication interface program. This way a network-independent access can be provided to the application layer.

We believe that in the near future the next generation networks will make it possible to use multiple network connections simultaneously. The concept of being always best connected (ABC) introduced by Gustafsson and Jonsson [5] supports our opinion. The ABC concept means that the users of mobile terminals have the opportunity of choosing the network access technology that is most suitable for them. According to the concept, users will be able to choose also the access provider. Thus, the selection of the network connection to be used is not limited to a certain network operator, but the user can use any network connection provided by any network operator. If the future networks provide this kind of an inter-operator roaming the operators will start competing more rigorously, from which the customers will benefit. Especially, if it was possible to change the network operator on the fly, it would enable new types of network access markets to be born.

There are many studies that consider the situation where there are multiple network connections available [1, 3, 6, 7, 8, 12]. In these studies the aim is to choose a network connection to be used in data transmission. We, however, use multiple network connections simultaneously and our aim is to select a network connection for each component of the data in such a way that the costs and the time used in transferring are minimized. The optimization method we propose in this paper can be used, for example, within the framework presented in [7] or within other similar frameworks. The framework proposed in [7] includes a virtual network interface that hides the actual physical channels among which the used network connection is selected. Below the virtual interface there are a policy manager and an interface manager which together make the selection. The model also requires one additional system component to monitor the availability of the network connections. In this proposal a user can control the policy manager and thereby affect the selection of the network connection.

In [3] a method based on multi-attribute decision making is presented to rank the networks available and to choose a network connection to be used in data transmission. We could use this kind of a method to select candidate networks to be considered in our algorithm. The selected network connections would be such that it is possible to use them to transfer data when we consider the properties of both the mobile terminal and the networks available as well as the quality of service required. In addition, we could disregard some network connections if the user finds them to be too expensive to use. This, however, is not the topic of this paper. From now on, we assume that some kind of a selection is made, and all the network connections we consider in the NCS problem are such that it is possible to transfer the data using them when considering the properties of the terminal and the quality of service required.

3. MATHEMATICAL MODEL

The NCS problem was first introduced in [10], and it can be defined as follows. We assume that a mobile terminal is able to use several different kinds of network connections simultaneously. The connections are used to transfer data that consists of separate components that can be transferred independently. This means that the components can be transferred using different networks at the same time. The problem is to decide which network connection should be used for transferring each data component. The aim is to minimize the costs and the time used in transferring.

In this section, we present a mathematical model of the NCS problem originally formulated in [10]. Let us assume that there are m network connections available in a mobile terminal and data consisting of n separate components is to be transferred using the connections. The components are assumed to be independent of each other so that it is possible to send them in any order. All the m network connections are assumed to be such that it is possible to transfer the data using them. In other words, the quality of service properties of these connections are suitable for the data transmission in question.

Let d_{ij}, $i = 1, \ldots, m$, $j = 1, \ldots, n$, denote the time that is needed in transferring component j using connection i, and let c_{ij} be the cost of using connection i to transfer component j. We define that a binary variable x_{ij} equals one, when component j is transferred using connection i, otherwise it is zero. Now, the NCS problem can be stated mathematically as follows:

minimize

$$f_1(x) = \max_{i=1,\ldots,m} \sum_{j=1}^{n} d_{ij} x_{ij} \quad \text{and} \quad f_2(x) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

subject to

$$\sum_{i=1}^{m} x_{ij} = 1, \quad \text{for all } j = 1, \ldots, n, \qquad (1)$$

$$x_{ij} \in \{0, 1\}, \quad \text{for all } i = 1, \ldots, m \text{ and } j = 1, \ldots, n. \qquad (2)$$

The objective function f_1 denotes the time used in transferring the components, that is, the connectivity time of the connection used for the longest time interval. The objective function f_2 expresses the costs of the transfers. The constraints (1) require that each component is transferred using exactly one connection. The constraints (1) and (2) form a set of feasible solutions S, from which the best solution to the problem is searched.

When the problem is solved, the solution tells which connection is used for transferring each component. It does not pay attention to the order in which the components are transferred on each connection because neither the time used in transfers nor the total costs depend on the order of the components. The components can be ordered on each connection according to some criteria. They can be transferred, for example, in the increasing order of their sizes. Note that in this model, we do not define how the cost of transferring a component is calculated but only assume that the cost c_{ij} can be expressed for each component j and each connection i. The cost c_{ij} can also be an estimate of the actual cost as long as the estimates for different connections are comparable.

4. SOLVING THE PROBLEM

Our aim is to develop an automatic method to solve the NCS problem fast. In addition, the method can use only little computational capacity so that it is possible to implement it as a middleware application in a mobile terminal. Since we have two objectives, we need a method that quickly and automatically obtains a good compromise between the objectives. In [10], we analyzed the main characteristics of the NCS problem and compared different multiobjective optimization methods for solving it. We found the following method to be the most suitable for our purposes. The method produces a compromise solution by minimizing the following function

$$\max_{i=1,2} w_i \frac{f_i(x) - z_i^{mid}}{z_i^{nad} - z_i^*} + \rho \sum_{i=1}^{2} w_i \frac{f_i(x) - z_i^{mid}}{z_i^{nad} - z_i^*} \quad (3)$$

subject to $x \in S$, where z_i^{mid} is a middle point located in the middle of the ranges of the objective function, that is,

$$z_i^{mid} = \frac{z_i^{nad} + z_i^*}{2}$$

for $i = 1, 2$, and z^* and z^{nad} are the ideal and nadir objective vectors representing the best and worst objective function values, respectively. The coefficient ρ is a small positive scalar, and the ratio of the positive weighting coefficients w_1 and w_2 represents the rate at which the user of the mobile terminal is willing to trade off values of the objective functions. We use value 0.001 for the coefficient ρ and values 1 and 2 for the weighting coefficients w_1 and w_2, respectively. These values were used also in [10].

Before minimizing the objective function (3) the ideal and nadir objective vectors have to be calculated. The *ideal objective vector* z^* that represents the best values of the objective functions can be obtained by minimizing each objective function separately subject to $x \in S$. The *nadir objective vector* z^{nad} representing the worst values of the objective functions can be obtained at the same time the ideal objective vector is calculated since we consider only two objective functions [4]. Calculating the ideal and nadir objective

vectors means additional computations slowing down the solution method considerably. This can be avoided by using their approximations because they are mainly needed for scaling purposes. We use vector $(0, 0)^T$ to approximate the ideal objective vector. The nadir objective vector is estimated from the problem data as follows. The first component of the vector is the objective function value $f_1(x)$ related to a solution x where every component is transferred using the slowest connection. The second component of the vector is the objective function value $f_2(x)$ related to a solution x where every component is transferred using the most expensive connection. These estimates may be very rough. However, the estimates are sufficient for our purposes [10].

Minimizing the objective function (3) is computationally demanding in the case of the NCS problem because of the integer variables [11]. In other words, it is time-consuming to solve large problem instances using exact methods. In addition, many exact methods require a lot of computational capacity to solve large integer programming problems. In mobile terminals, there is not much computational capacity available for this kind of optimization, and the users of mobile terminals are rarely prepared to wait more than a very short time. Therefore, we have developed a heuristic to solve the NCS problem [11]. The heuristic minimizes the objective function (3) approximately, achieving a good compromise solution. The heuristic is fast and, in addition, it does not require a lot of computational capacity, which makes it possible to implement it in a mobile terminal. The heuristic can be summed up as follows: (For more information on the heuristic, see [11].)

1. **Initial solution**. For each component $j = 1, \ldots, n$, find the network connection that gives the lowest value of the objective function (3) for the current partial problem, and assign the component to the connection. The current *partial problem* consists of the component j and the components that are already assigned (with fixed assignments). All the connections are included in the partial problem, and the ideal and nadir objective vectors are approximated for each partial problem separately.

2. **Improvements**. Set the initial solution as the current solution. Repeat twice: For each component $j = 1, \ldots, n$, search for the solution having the lowest value of the objective function (3) among the solutions in which the component j is transferred using another link instead of the link used in the current solution and the assignments of the other components are fixed. Set the new solution found as the current solution if the value of the function (3) is lower than in the current solution.

3. **Ordering**. Order the components assigned to each network connection according to some criteria, for example, the size. The components will be transferred in that order.

Note that originally in [11] the improvement phase in Step 2 is repeated until a stopping criterion (not explicitly defined) is satisfied. Because of the computational results presented in [11], we have decided to repeat the improvement phase twice. As mentioned earlier, the order in which the components are transferred has no impact on the time that

it takes to transfer the components nor on the costs of the transfer. Therefore, we can ignore the order when assigning the components to the connections and order the components afterwards in Step 3. The components need to be ordered to obtain a *transmission schedule* which is followed during the data transmission.

5. DYNAMIC ENVIRONMENT

In the previous section, we briefly presented a heuristic for solving the NCS problem. The heuristic was developed assuming a static environment where the properties of the connections do not change during the transmission. This kind of an assumption can be made only when the transmission time is short. When the transmission rates and the prices change during the transmission, a solution given by the heuristic may need revising. In addition, a network connection can be lost during the transmission if, for example, the mobile terminal enters an area that is not covered by that network. Then, the solution given by the heuristic may become infeasible and changing the solution is necessary in order to finish the transmission.

In order to cope with a dynamic environment the heuristic needs a rescheduling phase that improves the transmission schedule when possible during the transmission. The rescheduling phase takes into account changes in the transmission rates and in the pricing. In rescheduling, a component (or several components) is assigned to be transferred using another connection instead of the connection it is assigned to in the (current) schedule. We next describe the rescheduling phase in more detail.

When a change in the transmission rates or prices occurs, the amount of change is first controlled. If the change is larger than a prespecified limit B the rescheduling phase is run. The limit can be, for example, a certain percentage of the previous rate or price. The rescheduling phase is not run if the change in the transmission rates or prices is small because the solution rarely improves in this case. When there occur multiple changes at the same time the rescheduling phase is run if the largest change is large enough.

We also consider the case in which the network connections can be lost and later return. Now, we first need to control if a connection is lost or a new connection has appeared. If this is the case, the rescheduling phase is run. However, now the rescheduling phase begins by assigning the components that are assigned to the lost connections to other connections. The components considered here are the components that have not yet been transferred. In addition, if the transmission of a component was interrupted when a connection was lost, this component will also be assigned to another connection in order to ensure that the transmission will be finished. The assignment of the components to new links is done the same way as the initial solution is formed in the static version of the heuristic: Each component in turn is assigned to the connection that gives the lowest value of the objective function (3).

The rescheduling phase searches for the best solution (with respect to the objective function (3)) to a *partial problem* consisting of all the connections currently available and the components that have not yet been transferred. These components are collected into a set C and the network connections that are currently available are included in a set N. In addition, a set D is used for the components that are assigned to lost connections and need to be reassigned to other connections. The rescheduling phase consists of the following steps:

1. **Initialization**. Set $C = \emptyset$, $D = \emptyset$ and $N = \emptyset$. Control which connections are currently available and add those connections to the set N. Add to the set C all the components that are waiting to be transferred using the connections in N. Remember the current assignment of the components to the connections. The partial problem consists of the connections in N and the components in C.

2. **Initial solution**. Form an initial solution to the partial problem:

   - If there are no lost connections, the initial solution is the current assignment of the components in C to the connections in N. Go to Step 3.

   - Otherwise, add all the components assigned to the lost connections in the set D and remove the components from the transfer queues of the connections. If a transfer of a component was interrupted when a connection was lost, remove also this component from the transfer queues and add it to the set D.

   - Form the initial solution. For each component j in D: Remove the component j from the set D and add it to the set C. Assign the component j to the network connection i in N that gives the lowest value of the objective function (3) to the partial problem consisting of the components in C and the connections in N when the component j is assigned to the connection i and the assignments of the other components in C are fixed.

3. **Rescheduling**. Set the initial solution as the current solution. Use rescheduling to each component j in C: Form a set of solutions where the component j is assigned to another connection instead of the connection to which it is assigned in the current solution. Select among the solutions the one having the lowest value of the objective function (3) to be a *rescheduling solution.* If the rescheduling solution has a lower value of the objective function (3) than the current solution, set the rescheduling solution as the current solution.

It should be noted that we do not order the components again when rescheduling. The order of the components on each connection stays the same with the exception that the components that are moved to other connections are put last in the transfer queues. This is done at the same time as the rescheduling and, therefore, we do not have to perform any ordering afterwards, which saves some computational time. Remember that the rescheduling phase is run during the transmission and it should be as fast as possible to avoid delaying the transfer.

Figure 1 summarizes how the heuristic responds to a change in the network conditions.

6. ILLUSTRATIVE EXAMPLE

Next we present an example of the NCS problem. We use the term *static heuristic* to refer to the heuristic without the rescheduling phase and the term *dynamic heuristic* to refer
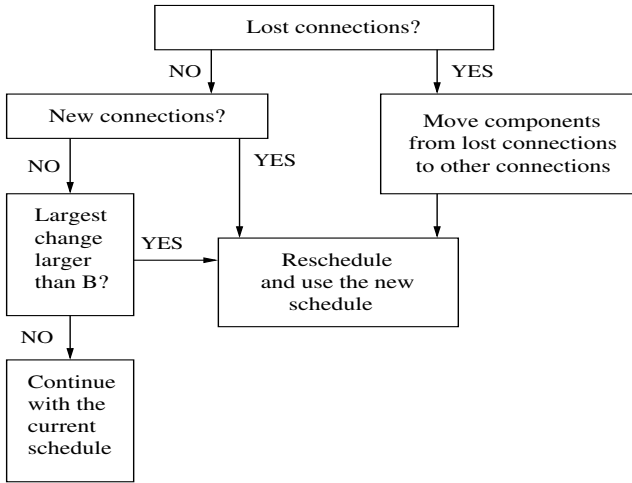
Figure 1: Responding to a change in the network conditions.

to the heuristic using the rescheduling phase. The problem instance we consider in this section has 86 components that form a real web page. The components are to be transferred using a set of eight network connections. The cost of transferring component j using network connection i is calculated using formula

$$c_{ij} = k_0 + k_1\sqrt{r_i} \cdot s_j \qquad (4)$$

where r_i is the rate of the connection i (Kbits per second), s_j is the size of the component j in bytes, and k_0 and k_1 are connection-specific coefficients. It should be noted that this formula for calculating the cost is only an example which shows that the price can consist of a fixed cost and a variable cost. How the cost is calculated has no effect on the heuristic.

The eight connections form three groups that have different parameter values in the pricing model. The groups represent competing operators that offer similar network connections with different prices. In the first group, the values for the parameters k_0 and k_1 are 300 and 1.1, respectively. In the second group the values are 100 and 2.3, and in the third group 30 and 0.8, respectively. The rates of the connections are 14.4, 59.2, and 384 Kbits per second in the first group, 14.4, 59.2, and 115 Kbits per second in the second group, and in the last group 14.4 and 59.2 Kbits per second. The cost calculated using the formula (4) is multiplied by 10^{-5} in order to have it in euros.

Let us now consider a dynamic network environment in which the rates of the network connections vary during the data transmission. The original rates of the network connections are assumed to be given by the operators of the networks and they present the maximum rates of the connections. The actual rates of the connections vary during the transmission. In addition, it is possible that a network connection is lost during a transmission due to, for example, movement of the mobile terminal. In our simulation, a change in the transmission rates occurs randomly with a mean of 0.7 changes in a second per network connection. When a change in the rate of a network connection occurs, the network connection is lost with a probability of 0.01. Otherwise, the rate of the connection is uniformly distributed between the maximum rate divided by three and

the maximum rate.

It should be noted that in this simulation example the pricing is fixed during the transmission and, therefore, the costs change only when the transmission schedule is changed. In addition, it should be noted that the number of changes occurring is large, which corresponds to a situation where the strength of the signal is weak or the mobile terminal is moving fast. This kind of a situation was chosen in order to demonstrate the performance of the dynamic heuristic.

Figures 2 and 3 illustrate the results of a simulation where the same data is transferred using two different strategies at the same time in the dynamic environment presented above. In one of the transmissions, the schedule given by the static heuristic is followed during the entire transmission. In the other transmission, the schedule is updated using rescheduling when a change in the transmission rates has occured and the change is larger than 20 percent of the previous rate. The times when a change occurs are marked by squares and circles in the figures. It should be noted that the transmission rates given to the static heuristic in the beginning of the simulation are the actual rates, not the maximum rates given by the operators. Using the maximum rates might result in a poor solution.

Figure 2 shows how the expected total time needed for transferring changes during the transmission. How the expected total costs change during the transmission is depicted in Figure 3. In this simulation example rescheduling makes the transmission faster but a bit more expensive. It should be noted that in this simulation the transmission using the static heuristic is stopped on a network connection during a period from 2.59 to 4.22 because the connection is not available then. (The other network connections are available and in use also during that period.) Therefore, in Figure 2 the total transmission time is not marked for that period (because it is not defined). The dynamic heuristic assigns the components waiting to be transferred on the lost connection to other connections using the rescheduling phase. Therefore, the transmission using the dynamic heuristic is not stopped during any period. This makes the total transmission time longer for the transmission using the static heuristic. However, this is only one reason why the dynamic heuristic makes the transmission faster. As we can see in Figure 2, the time needed for transferring using the static heuristic grows rapidly already in the beginning of the transmission. This is caused by a network connection that slows down considerably. The dynamic heuristic removes components from the connection keeping the transmission time shorter but at the same time making the transmission a bit more expensive.

This simulation example shows how the dynamic heuristic works. In this case, the dynamic heuristic makes the transmission shorter but a bit more expensive. In other simulation cases, the result may be the opposite or the dynamic heuristic may make the transmission both shorter and cheaper. Which is then better, to wait longer for the transmission to finish or pay little more for the transmission is a matter of preference. Our algorithm makes this decision automatically but, if wanted, user preferences can be taken into account in the method.

7. CONCLUSIONS

In this paper we have considered a system in which a mobile device can use multiple network connections simultane-
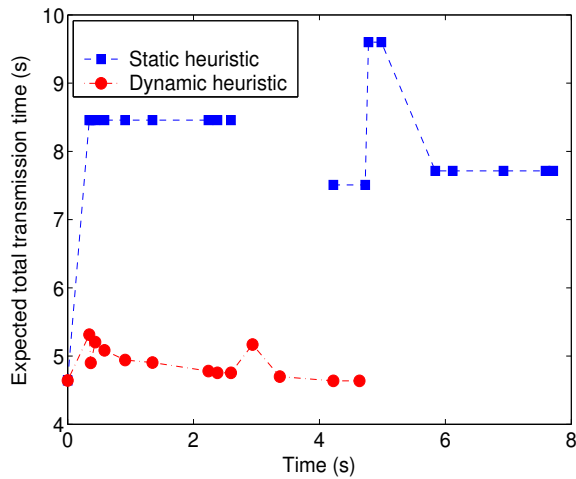
Figure 2: Example of how the expected total transmission time can vary during transmission.
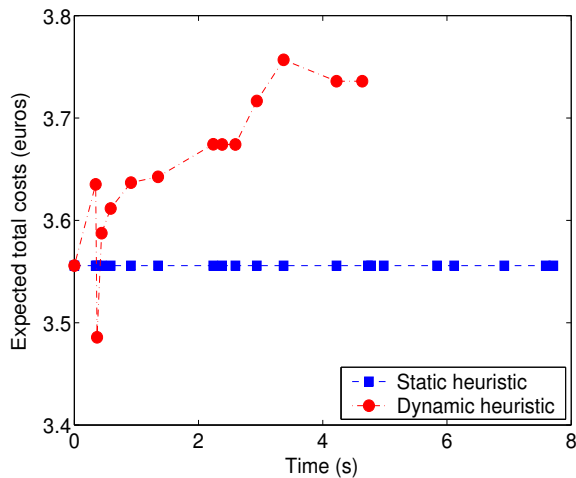

Figure 3: Example of how the expected total transmission costs can vary during transmission.

ously to transfer data. The problem is to select which network connection to use for each component of the data. This problem is called the network connection selection problem, and it has been formulated as a multiobjective optimization problem. To solve the problem, we have presented a simple but efficient heuristic that takes into account the dynamic network environment and improves the solution during the transmission when possible. The heuristic is fast and needs only little computational capacity and, therefore, it is possible to use it in mobile terminals during data transmission.

Future work will concentrate on adjusting the parameters of the heuristic using simulations. The main research question will be when it is profitable to use rescheduling. In other words, we need to determine when the change in the transmission rates and prices is large enough to make improving the current solution worthwhile.

8. ACKNOWLEDGEMENTS

9. REFERENCES

[1] E. Adamopoulou, K. Demestichas, A. Koutsorodi, and M. Theologou. Intelligent access network selection in heterogeneous networks - simulation results. In *Proc. of the 2nd International Symposium on Wireless Communication Systems*, pages 279–283, 2005.

[2] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty. NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, 50(13):2127–2159, 2006.

[3] F. Bari and V. C. M. Leung. Automated network selection in a heterogeneous wireless network environment. *IEEE Network*, 21(1):34–40, 2007.

[4] M. Ehrgott and D. Tenfelde-Podehl. Computation of ideal and Nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151(1):119–139, 2003.

[5] E. Gustafsson and A. Jonsson. Always best connected. *IEEE Wireless Communications*, 10(1):49–55, 2003.

[6] A. Iera, A. Molinaro, C. Campolo, and M. Amadeo. An access network selection algorithm dynamically adapted to user needs and preferences. In *Proc. of the 17th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2006.

[7] J. H. Li, S. Luo, S. Das, T. McAuley, M. Patel, A. Staikos, and M. Gerla. An integrated framework for seamless soft handoff in ad hoc networks. In *Proc. of the Military Communications Conference*, 2006.

[8] L. Liang, H. Wang, and P. Zhang. Net utility-based network selection scheme in CDMA cellular/WLAN integrated networks. In *Proc. of the IEEE Wireless Communications and Networking Conference*, pages 3313–3317, 2007.

[9] J. Mitola. Cognitive radio for flexible mobile multimedia communications. In *Proc. of the IEEE International Workshop on Mobile Multimedia Communications*, pages 3–10, 1999.

[10] A. Setämaa-Kärkkäinen, K. Miettinen, and J. Vuori. Best compromise solution for a new multiobjective scheduling problem. *Computers & Operations Research*, 33(8):2353–2368, 2006.

[11] A. Setämaa-Kärkkäinen, K. Miettinen, and J. Vuori. Heuristic for a new multiobjective scheduling problem. *Optimization Letters*, 1(3):213–225, 2007.

[12] Q. Song and A. Jamalipour. An adaptive quality-of-service network selection mechanism for heterogeneous mobile networks. *Wireless Communications and Mobile Computing*, 5(6):697–708, 2005.