Evolutionary Multiobjective Optimization for Base Station Transmitter Placement with Frequency Assignment

Nicole Weicker and Gabor Szabo and Karsten Weicker and Peter Widmayer

Abstract

We propose a new solution to the problem of positioning base station transmitters of a mobile phone network and assigning frequencies to the transmitters, both in an optimal way. Since an exact solution cannot be expected to run in polynomial time for all interesting versions of this problem (they are all NP-hard), our algorithm follows a heuristic approach based on the evolutionary paradigm. For this evolution to be efficient, that is at the same time goal-oriented and sufficiently random, problem specific knowledge is embedded in the operators. The problem requires both the minimization of the cost and of the channel interference. We examine and compare two standard multiobjective techniques and a new algorithm, the steady state evolutionary algorithm with Pareto tournaments (stEAPT). One major finding of the empirical investigation is a strong influence of the choice of the multiobjective selection method on the utility of the problem-specific recombination leading to a significant difference in the solution quality.

I. Motivation

The engineering and architecture of large cellular networks is a highly complicated task with substantial impact on the quality-of-service perceived by users, the cost incurred by the network providers, and environmental effects such as radio smog. Because so many different aspects are involved, the respective optimization problems are a proper object for multiobjective optimization and may serve as real-world benchmarks for multiobjective methods.

For all cellular network systems one major design step is selecting the locations for the base station transmitters (BST Location problem) and setting up optimal configurations such that coverage of the desired area with strong enough radio signals is high and deployment costs are low.

For Frequency Division/Time Division Multiple Access (FD/TDMA) systems a second design step is to allocate frequency channels to the cells. For GSM (Global System for Mobile Communications) systems, a fixed frequency spectrum is available. This spectrum is divided into a fixed number of channels. For a good quality-of-service, network providers should allocate to each cell enough channels to satisfy all simultaneous demands (calls). The channels should be assigned to the cells in such a way that interference with channels of neighbor cells or inside the same cell is low. This problem is called in the literature the fixed spectrum frequency channel assignment problem (FS-FAP).

BST-Location (BST-L) and the FAP problem are known to be NP-hard: The Minimum Set Cover problem can be reduced in polynomial time to the BST-L problem, and the FAP problem contains the Vertex Coloring Problem as a special case. For both problems, BST-L and FAP, several heuristic based approaches have been presented [1], [2], [3], [4], [7], [8], [11], [13]. For the BST Location problem which considers also interferences, one interesting practical approach was presented in [24]. The most recent result is by Galota et. al. [10], showing a polynomial time approximation scheme for one version of the BST-L problem. Also for a weighted coloring version of the FAP an optimization algorithm was presented recently [29] that works for the special case of series-parallel graphs.

Two separate optimization steps, BST-L followed by FAP, must be viewed critically—a solution of BST-L restricts the space of possible overall solutions considerably and might delimit the outcome of the FAP optimization. Usually an iterated two-phase procedure is chosen to approach a sufficient solution. However, it is usually difficult to feed back the results of the second phase into an optimization of the first phase again. This paper shows that with today's increased availability of computing power it has become practically feasible to address the cellular network design problems (BST-L and FS-FAP) with an integrated approach that considers the whole problem in a single optimization phase. Since the two separate optimization steps do influence each other for practical problem instances, we expect better overall results when they are integrated into a single design step although the search space is enlarged drastically. To our knowledge, there were only very few papers that addressed the integrated problem (e.g. [14]), and none of them pursues an evolutionary approach. We are interested in exploring the potential that evolutionary algorithms have to offer.

This work has been partially funded by project TH-46/02-1.

Gabor Szabo and Peter Widmayer are with the Institute of Theoretical Computer Science, ETH Zürich, 8092 Zürich, Switzerland.

Nicole Weicker and Karsten Weicker are with the Institute of Formal Methods in Computer Science, University of Stuttgart, 70511 Stuttgart, Germany.

For successfully coping with the enlarged search space, two design issues are of major importance concerning the evolutionary algorithm. First, the general concept of evolutionary computation must be tailored to match the abundance of constraints and objectives of the integrated problem for cellular network design in its full practical complexity. Second, the multiobjective character of the problem must be reflected in the selection strategy of the algorithm such that a sensible variety of solutions is offered, reflecting the tradeoff between cost and channel interferences. Our experimental results demonstrate that even though we cannot guarantee bounds on the worst-case behavior, our evolutionary approach can handle real problem instances.

In the next Section, we present our formalization for the integrated cellular network design problem. Section III outlines the design criteria used to incorporate problem knowledge into the evolutionary algorithm as well as the options to handle multiple objectives. The concrete realization of our evolutionary algorithm, tailored to the problem, follows in Section IV. We show some of the experimental results in Section V and finally conclude in Section VI.

II. PROBLEM

For our problem we use a real teletraffic matrix for the region of Zürich. It is given through statistical data about population, buildings and land type [23]. From now on, we refer to the teletraffic in a certain unit area as *demand*. A demand node in our model has a fixed location and carries a certain number of calls per time unit. This number can vary across demand nodes. We consider a service area in which a set of demand nodes with different locations and numbers of calls are given. The task is to place transmitters in the service area in such a way that all calls of the demand nodes can be served with as little interference as possible. The transmitters may be given different power (signal strength) and capacity (number of channels). The power of a transmitter, together with a wave propagation function, determines the region (called *cell*) in which the transmitter can serve calls. The capacity of a transmitter is the number of different frequency channels on which the transmitter can work simultaneously. Electromagnetic interference may be a problem in an area where two signals with the same frequency or with very similar frequencies are present. As the cells of the transmitters can overlap, interference can occur between different transmitters and within one transmitter's cell.

To summarize the problem, we want to determine locations of transmitters, and to assign powers and frequencies to them such that all calls can be served. We aim at minimizing two objectives, the cost for the transmitters and the interference.

Let us now make the problem technically precise.

Let $A = ((x_{\min}, y_{\min}), (x_{\max}, y_{\max}), res)$ denote a *service area*, where $(x_{\min}, y_{\min})$ is the lower left corner of a terrain, given in some standard geographical coordinate system, $(x_{\max}, y_{\max})$ is the upper right corner of the terrain, and $res \in \mathbb{N}$ is the resolution of the terrain.We limit ourselves to grid points, for all purposes. That is, a point (x, y) is said to be in A if $x = x_{\min} + i \cdot res$ and $y = y_{\min} + j \cdot res$ for some integers $0 \le i \le \frac{x_{\max} - x_{\min}}{res}$ and $0 \le j \le \frac{y_{\max} - y_{\min}}{res}$. Then we write $(x, y) \in A$.

Let $D = \{d_1, \ldots, d_n\}$ denote a finite set of *demand nodes*, given by their position inside the service area A and the number of *calls* they carry, i.e. $d_i = (pos_i, r_i)$, where $pos_i \in A$ is the position of demand node d_i and $r_i \in \mathbb{N}$ is the number of calls at d_i, $1 \le i \le n$.

Let $\mathcal{F} = \{f_1, \ldots, f_k\}$, $k \in \mathbb{N}$ denote a fixed size *frequency spectrum*, where f_i is a *frequency channel* that can be used for communication. Each of the channels has a unique label from an ordered set (i.e. $f_i < f_j$ or $f_j < f_i$ for $i \neq j$ and $i, j \in \{1, \ldots, k\}$).

Let T denote a finite set of possible *transmitter configurations*, where $t \in T$ with $t = (pow, cap, pos, F)$ with the transmitting power $pow \in [MinPow, MaxPow] \subset \mathbb{N}$, the capacity of the transmitter $cap \in [0, MaxCap] \subset \mathbb{N}$, the position of the transmitter $pos \in A$ inside the service area, and the set of channels assigned to the transmitter $F \subset \mathcal{F}$ where $|F| \le cap$.

For our simple model, a *wave propagation function* $wp : T \to \mathcal{P}(A)$ is a function of the power and the position of a transmitter $t = (pow, cap, pos, F)$, i.e. $wp(t) = f_{wp}(pow, pos)$, and returns a region $cell \subset A$. In the simple, planar grid interpretation of the service area, the cell is a circle, discretized to the grid, where the position of the transmitter t is the center of the $cell_t$ and the power determines the radius. $cell_t$ is called the *cell* of the transmitter t. Inside the cell, the signal of the transmitter is strong enough for communication. The cell is defined so that outside the cell, the signal is not strong enough to produce interference with signals of other transmitters.

A demand node $d \in D$ is *covered* if all its calls are satisfied. A call from d is *satisfied* if there is at least one selected transmitter $t = (pow, cap, pos, F)$ that has a sufficiently strong signal at d (i.e. $d \in cell_t$) and has a channel $f \in F$ assigned for this call. Then transmitter t is said to be a *server* for d. More than one selected transmitter can serve the same demand node, but not the same call.

There are two kinds of channel interference, the co-channel interference and the adjacent-channel interference.

- The *co-channel interference* occurs between neighboring cells that use a common frequency channel. In contrast to the formula given in [23] for the computation of co-channel interference, we use a discrete model for the interference. We count for each selected transmitter the *noisy* channels, where a channel allocated to a transmitter t is said to be noisy if it has

an overlapping region with another transmitter using the same channel, and in their overlapping region there is at least one demand node satisfied by the transmitter t.
$NC_{CC}(t)$ is the set of all noisy channels for a selected transmitter t from a co-channel interference point of view.
We consider this model for the co-channel interference because it captures the worst case scenario in real situations: when two customers in the overlapping region of two transmitters make their calls and both get the same frequency channel assigned from the different transmitters, they hear only noise.

- The adjacent-channel interference appears inside one or between neighboring cells using channels close to each other on the frequency spectrum. Like in most other examinations in the literature, we consider adjacent-channel interference only inside the same cell and only between adjacent channels. To avoid this type of interference, usually a sufficient frequency gap $g_{AC} \in \mathbb{N}$ is specified that must be kept between assigned frequency channels. Then for a selected transmitter $t \in T$ we call $\min\{f_j, f_l\}$ a noisy channel from an adjacent-channel interference point of view, if $\exists\, f_j, f_l \in F,\ j \neq l,\ |f_j - f_l| < g_{AC}$.
$NC_{AC}(t)$ is the set of all noisy channels for a selected transmitter t from an adjacent-channel interference point of view.

Figure 1 gives a simple example for interference, where for transmitter A channels $1, 4, 10$ are noisy channels from a co-channel interference point of view since B uses the same channels and A serves demands in the overlapping region. With a frequency gap $g_{AC} = 2$, channels $3, 4, 9$ are noisy from an adjacent-channel interference point of view since channels $4, 5, 10$ are used too and lie within the frequency gap.

The set of all *candidate* solutions $\mathcal{S}$ is defined as

$$\mathcal{S} \;=\; \left\{ \{t_1, \ldots, t_k\} \;\middle|\; k \in \mathbb{N} \text{ and } t_i \in T \text{ for all } 1 \leq i \leq k \right\}.$$

The set of all *feasible* solutions $\mathcal{F}$ with respect to a set of demand nodes D is defined as

$$\mathcal{F} \;=\; \left\{ S \in \mathcal{S} \;\middle|\; \text{all } d_i \in D \text{ are covered by } S \right\}.$$

The *cost of a feasible solution* $S \in \mathcal{F}$ is computed by $cost(S) = \sum_{i=1}^{|S|} cost(t_i)$, where the *cost of one transmitter* $t_i = (pow_i, cap_i, pos_i, F_i)$ is a function $cost : T \to \mathbb{R}^+$ of the power and the capacity of the transmitter that is monotonous in both parameters: $cost(t_i) = f_c(pow_i, cap_i)$.

The *interference ratio of a feasible solution* is

$$IR(S) = \frac{\sum_{i=1}^{|S|} |NC(t_i)|}{\sum_{i=1}^{n} r_i},$$

where $NC(t_i) = NC_{CC}(t_i) \cup NC_{AC}(t_i)$.

The *goal* of the optimization is to find a feasible solution $S \in \mathcal{F}$ s.t. $cost(S)$ and $IR(S)$ are as small as possible.

III. Design criteria and multiobjective methods

The necessity to tailor an evolutionary algorithm to a specific problem is not only a conclusion of the *No Free Lunch Theorems* of Wolpert and Macready [27] but has also been applied many times by practitioners (see e.g. [5]). However, few general guidelines are available for designing an algorithm. In case of the base station transmitter placement with frequency assignment, the following design criteria have been regarded through the whole process of incorporating domain knowledge.

1) First of all, it has to be guaranteed that the representation is able to express all candidate solutions.
2) Since the problem includes certain constraints, it has to be guaranteed that any individual that can be produced by the genetic operators presents a feasible candidate solution, or, if this cannot be guaranteed, can be repaired to a feasible one.
3) Next, every point in the solution space should be reachable by the evolutionary operators at any step of the evolutionary algorithm. Each operator must possess its reverse.
4) The evolutionary operators have to be chosen in such a way that a balance between exploration and exploitation of the search space can be reached. This means that there is a need not only for problem specific operators guaranteeing only little changes to an individual, but also for randomly driven operators able to explore new areas of the search space.

One of the difficulties of the considered problem (see Section II) lies in the combination of three aims: we want to cover all demand nodes, with minimal costs for the needed transmitters and also minimal interference. The first aim is a constraint and, thus, we decided to use a genetic repair approach [28], i.e. in all candidate solutions all demand nodes must be covered—this is enforced using a repair function (see Section IV-B).

When defining genetic operators, the need for the repair function should be as small as possible to have a high correlation between parents and offspring. Moreover, we want to reach a good combination of directed search operators (resulting in an exploitation) and those which work more randomly (bringing the necessary exploration component). Also we want to use the power of recombination operators that combine different solutions in a meaningful way. The operators are presented in Sections IV-D and IV-E.

For the handling of the two minimization objectives a vast variety of multiobjective methods is available. All those algorithms are used to produce potentially optimal candidate solutions as elements of the Pareto front. At any stage of the evolution, the Pareto front is the set of current candidate solutions that are non-dominated by other current candidate solutions, i.e. there is no other feasible solution available currently that will yield an improvement in one objective without causing a degradation in at least one other objective (as introduced by Pareto [16]). An early approach has been the vector evaluated genetic algorithm (VEGA) [19] that often produces solutions distinct from the Pareto set or even favors rather extreme candidate solutions. Another rather intuitive method is the use of aggregating functions like in a weighted sum [21]. However the projection of multiple objective values to one scalar value handicaps concave regions of the Pareto front during search [17]. In the 1990s, research concentrated primarily on methods using the Pareto dominance directly. Examples are the Niched-Pareto Genetic Algorithm (NPGA) [12], the Nondominated Sorting Genetic Algorithm (NSGA) [20], and the Multi-Objective Genetic Algorithm (MOGA) [9]. However, the price for the success of these technique has been high complexity for sharing to spread the solutions across the Pareto front and to check for Pareto dominance. Also those algorithms lack a technique for elitism. Therefore, several algorithms have been proposed recently that tend to avoid those shortcomings, e.g. the Strength Pareto Evolutionary Algorithm (SPEA) [31], the improved SPEA2 [30], the improved NSGA-II [6], and the Multi-Objective Messy Genetic Algorithm (MOMGA) [25]. We decided to compare SPEA2, NSGA-II, and a new multiobjective steady state algorithm described in Section IV-F.

IV. CONCRETE REALIZATION

A. Representation

In order to represent a candidate solution within the evolutionary algorithm, we decided to use the native representation inherent in the problem description of Section II. That means each individual is of the form $ind \in \mathcal{S}$. Note that this is a variable length representation since the number of transmitters is not fixed.

An individual that represents a feasible candidate solution $ind \in \mathcal{F}$ is called a *legal individual*.

B. Repair function

An individual that has been created and is illegal at an intermediate stage of the algorithm can be transformed to a legal individual by means of a so-called repair function.

To repair an individual we traverse the demand nodes in any order. If a demand is not totally covered, then the appropriate one of the following actions is taken:

1) If there exist transmitters whose cells cover the demand node and the transmitters have free capacity to satisfy the unsatisfied calls, then the one with strongest signal will be selected to satisfy the calls. For this transmitter new channels are allocated for the unsatisfied calls.
2) If there exists no transmitter with free capacity whose cell covers the demand node, then for one of the neighboring non-maximum capacity transmitters, power is increased to cover the demand node. Which of the neighboring transmitters will be changed, is decided based on the extra deployment cost introduced by this change. The one with a minimum cost change will be chosen to satisfy the calls. If the minimum cost change will be bigger than the cost of introducing a new transmitter with some default configuration that can satisfy the calls or each neighboring transmitter already operates at maximum capacity, the action in step 3. is taken instead.
3) If none of the above actions are possible, then a new transmitter is introduced having the same or a neighboring location with the demand node in focus. This transmitter gets default power and also default capacity. The frequency channels will be allocated for the unsatisfied calls of the demand node.

In the repair operator we consider only the deployment cost as a criterion to decide which repair action will be chosen. This repair function results always in a legal individual.

C. Initialization

To initialize individuals at random, we start with an empty individual, and we fill it with transmitters by applying the repair function. To produce another, different, individual, we just reorder the sequence of demands, and then repeat the first step. The reason for the reordering lies in the property of the repair function to take the order into account.

This procedure has the advantage of producing legal individuals only. A pure random setting of the single values of an individual would instead lead with a high probability to an illegal individual.

It is perhaps a small disadvantage of this approach that the maximal population size will depend on the number of demand nodes, since the described initialization will give us only as many different individuals as there are permutations of the sequence of demand nodes. However this is not an issue for our real-world problem instances.

D. Mutation

Just like in most successful real-world applications of evolutionary algorithms, we need to include problem knowledge in the genetic operators to make the overall process effective and efficient. Since there are some rules of thumb used by experts to get better solutions, we introduce several mutation operators that use information produced by the evaluation function. These mutation operators are able to yield local changes of a given solution. We call them *directed mutations*. Operators with a similar intention have also been used in time tabling (e.g. [18]).

But only using such operators cannot guarantee that all points in the search space are reachable. Therefore we will also introduce some additional mutation operators that do not consider problem knowledge. We will call them *random mutations*.

Some of the directed and random mutation operators will change the individual in such a way that we cannot guarantee the individual to be still legal. These operators are *DM2*, *DM3*, *DM6*, *RM1*, *RM3*, *RM4*. Then, the repair function has to be applied.

Directed mutations use additional information produced; their application is limited to situations that satisfy certain preconditions.

DM1	Precondition	There exist transmitters with unused frequency channels.
	Action	Reduce the capacity.
	Comment	The goal is to reduce cost.
DM2	Precondition	There exist transmitters with maximal capacity that use all frequency channels.
	Action	Place a transmitter with default power and capacity in the neighborhood.
	Comment	The goal is to introduce micro-cells in areas with high number of calls.
DM3	Precondition	There exist transmitters with big overlapping regions.
	Action	Remove such a transmitter.
	Comment	The goal is to reduce the interference by reducing the overlapping regions.
DM4	Precondition	There exist transmitters with big overlapping regions.
	Action	Decrease the power of such a transmitter in a way that all satisfied calls remain satisfied.
	Comment	The goal is to reduce cost and interference.
DM5	Precondition	Interference occurs.
	Action	Change one of a pair of interfering frequency channels.
	Comment	The goal is to reduce interference.
DM6	Precondition	There exist transmitters satisfying only a small number of calls.
	Action	Delete such a transmitter.
	Comment	The goal is to reduce cost.

Pure random mutations can be applied to any individual; there are no preconditions.

RM1	Action	Change the position of a randomly chosen transmitter leaving power and capacity unchanged.
	Comment	This operator is needed because the placements of the transmitters in the service area are not randomly chosen.
RM2	Action	Introduce a new randomly generated individual.
	Comment	This is done by applying the repair function starting with an empty individual as described in Section IV-C. A random permutation of the demand node sequence is used. This operator alone does not guarantee the reachability of all points in the search space, since the repair function follows some strict rules. But with this mutation it is possible to bring fresh genetic material into the optimization.
RM3	Action	Change randomly the power of one randomly chosen transmitter.
	Comment	This operator is necessary to keep a balance to the directed mutation *DM4*.
RM4	Action	Change randomly the capacity of one randomly chosen transmitter.
	Comment	This operator is necessary to keep a balance to the directed mutation *DM1*.
RM5	Action	Change randomly the frequency channels allocated by one randomly chosen transmitter.
	Comment	This operator is necessary to keep a balance to the directed mutation *DM5*.

E. Recombination

Additionally to the different mutation operators, we want to use the possibility of combining genetic material of two individuals. Such a recombination makes the most sense if we also include problem knowledge, so that the probability of combining good characteristics of the parents is high.

The problem at hand has the characteristic that it is possible to evaluate an individual according to parts of the terrain (i.e. parts of the demand list). The aim of the recombination operator is to take good parts from the parents and to merge them for constructing a new individual – the offspring.

Our recombination operator is based on a decomposition of the service area (terrain) into two halves along one of the dimensions (vertical or horizontal). For each half we evaluate the fitness of the parent individuals, and the offspring will inherit the configuration for each of the sub-areas from the parent that was more fit for that sub-area.

With this approach, there might be some undesired effects close to the cutting line of the service area. If the offspring inherits from the parents the transmitters that were located close to the cutting line, huge overlapping regions can occur which probably lead to high interference. To avoid such undesired effects, we leave a margin of the size of a maximum cell radius on both sides of the cutting line, and we inherit the transmitter configurations from the parents only for the reduced half regions. In Figure 2 an example of the recombination operator can be seen.

The recombination operator may lead to illegal individuals that have to be repaired by the repair function.

F. Selection

The selection method is based on the cost $cost(ind)$ and the interference $IR(ind)$ of the individuals ind in the population.

We have investigated the two standard multiobjective methods Strength Pareto Evolutionary Algorithm 2 (SPEA2) [30] and the fast elitist Nondominated Sorting Genetic Algorithm NSGA-II [6].

SPEA2 uses an external archive where the best candidate solutions are stored. Each individual in the archive and the population gets a strength value assigned which denotes how many individuals it dominates. The raw fitness of an individual results as the sum of the strength values of all individuals that dominate the individual. The final fitness is obtained by adding a density information to the raw fitness which favors individuals with fewer neighbors from a set of individuals with equal raw fitness. The parental selection is implemented as a tournament according to the final fitness values. The time complexity of the algorithm depends primarily on the density computation needed for the integration phase. If both the archive size and the population size are N the integration of N newly created individuals takes $\mathcal{O}(N^2)$. If the archive size is N and the population size is 1, the time to integrate N individuals is $\mathcal{O}(N^3)$.

NSGA-II computes all layers of non-dominated fronts. This results in a rank for each individual. Furthermore a crowding distance is computed that is a measure for how close the nearest neighbors are. Selection takes place by using a partial order where an individual with lower rank is considered better and if the individuals have equal rank the individual with bigger crowding distance is preferred. The complexity of this method is determined by the expensive computation of the non-dominated fronts; it turns out to be $\mathcal{O}(mN^2)$, where m is the number of objectives and N is the population size.

Both standard methods have a rather generational character. SPEA2 may be used in a steady state mode but the time complexity increases considerably. Also, there are no recommendations concerning the population and archive sizes in the primary literature.

At a very recent stage of our application it was decided that a steady state approach might be interesting, since it enters new individuals immediately and favors a faster evolution speed. This is of big importance since the consideration of more complex wave propagation models may require much more time per evaluation. Therefore, a new steady state selection was developed that pays special attention to the replacement strategy and its time complexity. We refer to the algorithm as steady state evolutionary algorithm with Pareto tournaments (stEAPT).

Both the parental selection as well as the replacement strategy are based on a ranking strategy that takes into consideration the concept of domination. We consider two subsets of the population when assigning a rank (fitness value) to an individual ind, namely

- $Dominates(ind)$, the set of individuals that are dominated by ind, and
- $IsDominated(ind)$, the set of individuals that dominate ind.

The population is stored in a two-dimensional range tree, where the keys are the two objectives. We use the two-dimensional dictionary data structure from the Library for Efficient Data structures and Algorithms (LEDA) [15]. This data structure can handle two dimensional range queries in time $\mathcal{O}(k + \log^2 n)$ where k is the size of the returned set and n is the current size of the dictionary. Insert, delete, and lookup operations take time $\mathcal{O}(\log^2 n)$.

As a scalar fitness to be minimized the following value

$$fitness(ind) = |IsDominated(ind)| \cdot PopSize + |Dominates(ind)|,$$

is assigned to each individual. Clearly the number of dominating individuals establishes a primary ranking in the population. This is the primary cause for the selective pressure towards the overall Pareto front. If there are individuals that are dominated by an equal number of individuals, those are preferred that dominate fewer individuals. This is primitive mechanism to favor individuals from less crowded regions in the objective space. Apparently this is based on the assumption that a considerable fraction of individuals is dominated. If the complete population consists of non-dominated individuals selection acts as a mere uniform selection and genetic drift may occur. However, in all experiments using the real-world problem this appeared never to be a problem.

The parental selection is implemented as a tournament selection using the fitness value. After the variation operations a new individual new has to be integrated into the population. For this purpose the sets $IsDominated(new)$ and $Dominates(new)$ are computed. Now three different cases must be distinguished.

Case 1: Both sets are empty. That means that new is a new non-dominated candidate solution and should be inserted into the population. The worst individual ind according to the fitness value is deleted and the rank of all individuals that are in $IsDominated(ind)$ is decreased by one (light gray area in Figure 3). The new individual new is inserted with rank 0.

Case 2: The set $Dominates(new)$ is not empty, i.e. there is at least one individual that is worse than new according to the Pareto dominance. Therefore, the worst individual $ind \in Dominates(new)$ is deleted. Individual new is inserted with its properly computed rank. The rank of all individuals in $IsDominated(ind) \setminus IsDominated(new)$ is decreased by one (light gray area in Figure 4). For individuals in $Dominates(new)$ the rank is increased by $PopSize$.

Case 3: The set $Dominates(new)$ is empty and the set $IsDominated(new)$ is not empty. The new individual appears to be no improvement over any individual in the population and is therefore dropped. The case is shown in Figure 5.

For the case that all individuals are in the Pareto front, we considered to compute a crowding measure online to determine the worst individual and to prevent genetic drift. However, we did not encounter such a situation so far.

Algorithm 1 stEAPT algorithm

INPUT: population size N, maximum number of generations G
Initialization: create initial population by using the repair function, construct a 2d range tree, with cost and interference as keys; $t \leftarrow 0$
Evaluation and fitness assignment for the population
while $t \leq G$ **do**
  Parental selection: perform tournament selection
  Variation: create one new individual by using one of the evolutionary operators
  Evaluation: compute objectives for the new individual
  Replacement: integrate new individual and update range tree data structure and rankings
end while
OUTPUT: non-dominated set (current Pareto front)

G. *Algorithm*

The main loop of stEAPT is sketched in Algorithm 1; it follows the usual steady state scheme. For the variation of a selected individual only one of the operators, directed mutation, random mutation, or recombination, is applied. The probability for the application of these different kinds of operators is set by the parameter p_{DM} for the probability to apply a directed mutation and p_{RM} for the probability to apply a random mutation. $p_{DM} + p_{RM} < 1$ is required and the probability for applying recombination follows as $p_{RC} = 1 - p_{DM} - p_{RM}$. The repair function is applied to each newly created illegal individual.

V. Experiments

A. *Experimental setup*

The three described multiobjective methods are applied to realistic demand distributions on a $9000 \times 9000m^2$ area in the city of Zürich. We use two different resolution grids on top of the service area. The demand nodes are distributed on a $500m$ resolution grid, where for the transmitter positions we use a finer grain resolution of $100m$.

The number of calls at a certain demand node are computed according to the formula described in [23]. This formula is based on information related to the call attempt rate (considered the same for every mobile phone), number of mobile units per square kilometer, the mean call duration and the size of the area represented by the demand node. All these entities relate factors like land usage, population density and vehicular traffic with the calling behavior of the mobile units.

For the studied service area we have $|D| = 288$ demand nodes with a total number of 505 calls. In Figure 6 we can see the demand distribution: each circle represents a demand with its radius proportional to the number of carried calls. The empty area is lake Zürich.

We have chosen a maximum number $|F| = 128$ of frequency channels, to closely reflect reality for the GSM900 systems. The adjacent-channel gap that should be respected at each cell in order to have no adjacent-channel interference was chosen to be $g_{AC} = 2$. The maximum capacity of a transmitter is derived from the maximum number of channels and the adjacent-channel gap: $MaxCap := \frac{|F|}{g_{AC}} = 64$ in our case. We use a simple isotropic wave propagation model, where propagation depends only on the transmitting power (in dBmW). The cell radius of a transmitter is computed as $wp(t_i) := pow_i \cdot 25$. We have a discrete set of power values that can be set for the transmitters: $MinPow = 10$, $MaxPow = 130$ in dBmW, with increments of 1 dBmW. The transmitters choose among a discrete set of positions, given by the terrain resolution and the service area. The deployment cost of each transmitter is $cost(t_i) := 10 \cdot pow_i + cap_i$, for $t_i = (pow_i, cap_i, pos_i, F_i)$.

The three multiobjective methods have been applied to the given problem using various settings for population size, tournament size, number of objective evaluations, and probability distribution for the different types of operators (p_{DM}, p_{RM}, p_{RC}). For each algorithm with a specific parameter setting 16 independent, consecutive runs have been executed.

B. Statistical comparison

Single runs could identify several candidate solutions on a very high competitive level which justifies the general approach of this project. But there is still a very high variance in the quality of the runs using the same method and parameters. Although the development of very specialized operators was emphasized, the problem landscape seems still to be so rugged that there are many local optima where the non-dominated set of candidate solutions can get stuck.

As a consequence the comparison of different parameter settings or multiobjective methods turns out to be difficult. However to get statistical confidence concerning our conclusions we have chosen the following approach. Given a Pareto front of one run and the observation that the front is convex in almost all experiments, one candidate solution may be chosen using a weighted sum. If this is done for all runs of two different algorithms, we get two data series consisting of 16 weighted sums representing the best solution concerning the respective weights. Those data series may be compared using Student's t-test to get an idea whether there is a significant difference. In particular the two objectives are scaled, $\widehat{IR}(ind) = \frac{IR(ind)}{0.7}$ and $\widehat{cost}(ind) = \frac{cost(ind)-7500}{4500}$, and the weighted sum $WS(ind) = \alpha\widehat{IR}(ind) + (1-\alpha)\widehat{cost}(ind)$ is considered with $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

This comparison turns out to be very useful for the optimization problem at hand since it reduces the visually hardly interpretable data to exact numbers. In the comparisons given in the following two sections, all values of α support the respective statements.

C. Parameter settings

In our experiments, there are too many possible parameter settings to try them all. Unfortunately, there is no way to know the "best" parameter values right from the start. As an unavoidable consequence, a set of experiments cannot yield an insight that we can claim in full generality. Nevertheless, after plenty of experiments, we gained a "feeling" for what would be parameter settings that reveal an underlying principle.

The examination of a wide range of population sizes lead finally to the population size 80 that is used in all experiments. Concerning SPEA2 both the archive size and the population size are chosen to be 80. In the same way the tournament size was chosen to be 5. Also a set of experiments with varying number of evaluations has led to the judgment that 64'000 evaluations are sufficient since no substantial improvement takes place after that in most experiments.

Concerning the probability distribution for the different operators, experiments show that the problem specific operators alone ($p_{DM} = 1$ or $p_{RC} = 1$) lead to a quick convergence of the population on a bad quality level for both objectives. This is reasonable since these special operators are not able to reach every point in the search space and tend to get stuck in local optima. The details of these experiments are not included in this article.

The random mutations alone yield fairly good results, as can be seen in Figures 7, 8, and 9 which show the non-dominated individuals of the 16 runs.

As we can see in Figures 10, 11, and 12 the combination of random and directed mutations gives better results. The probabilities $p_{DM} = p_{RM} = 0.5$ are used. The difference is significant according to the statistical investigation described above. The interaction between the explorative global (random) and the exploitative local (directed) operators provides the power to find better solutions than with random mutation alone. This is also in accordance with the theoretical findings of Weicker [26] concerning the shift from exploration to exploitation during search.

Various probability distributions putting different emphasis on the three types of operators have been investigated. The best results could be obtained using $p_{DM} = 0.3$, $p_{RM} = 0.3$, and $p_{RC} = 0.4$. Again the difference to the two previously described probability settings are significant. The results are displayed in Figures 13, 14, and 15.

D. *Multiobjective methods*

The comparison of the three presented multiobjective methods turns out to be difficult. A visual comparison of Figures 7–15 leads to no result. The statistical comparison shows also that no significant difference can be proven for the two operator schemes using no recombination. In fact for all considered values of α, the average weighted sums are almost identical across all three multiobjective techniques.

However, concerning the best performing scheme with $p_{DM} = 0.3$, $p_{RM} = 0.3$, and $p_{RC} = 0.4$ shows a different result. The new method stEAPT reaches worse average weighted sums for all values of α. The best values are reached by NSGA-II for $\alpha = \{0.3, 0.4, \ldots, 0.9\}$. And for $\alpha = 0.6$ and $\alpha = 0.7$ the difference between NSGA-II and stEAPT turns out to be significant with an error probability of less than 0.06.

These findings show that there is no difference in the performance of the three compared methods concerning the hard real-world problem as long as the mutation operators are considered. Here the stEAPT method proves to be a useful alternative especially since the computation time is considerably smaller than in the other approaches. But as soon as a very specialized recombinative operator is added, the choice of the multiobjective method makes a difference. The only possible explanation for this phenomenon can be, that both NSGA-II and SPEA2 select the parental individuals in a way that is more appealing to the recombination. As a consequence better offspring are produced. Presumably, the more sophisticated crowding procedures within the fitness computation of NSGA-II and SPEA2 lead to a more diverse recombinative behavior.

VI. Conclusion and future work

In this paper, we have demonstrated that evolutionary algorithms are a strong enough tool to tackle the real-world problem of base station transmitter placement and frequency assignment, based on the real situation for the region of Zürich, with all its complications, in the objectives as well as in the constraints. The success of the evolutionary approach is primarily due to the tailored problem specific operators.

As a new multiobjective technique the stEAPT approach is introduced which combines a steady state scheme with a very efficient data structure leading to superior time complexity. In general the stEAPT algorithm proves to be competitive to NSGA-II and SPEA2—at least for the considered hard real-world problem.

However, the probably most intriguing outcome of this examination is the interplay between the crowding mechanisms and the recombination operator. Here NSGA-II and SPEA2 turn out to support the recombinative potential of the population in a considerably better way than the simpler stEAPT.

In the future, we plan to do the following.

1) Concerning the real-world application, a rational valued problem instance will be considered and compared to the results with the coarse discrete grid. Also the minimization of electro smog will be considered as an additional objective. A more sophisticated wave propagation model is planned too that takes the environment into account.
2) The phenomenon of crowding and recombination is to be investigated in more depth. Probably one approach could be the statistical analysis of the behavior of the different genetic operators. This could also lead to an adaptation scheme for the operator application probabilities over time.
3) The stEAPT method needs to be investigated on traditional benchmark functions too. Also the incorporation of an efficient crowding mechanism for parental selection should be one possible focus to improve the technique.

Acknowledgement

We gratefully acknowledge discussions with Konrad Schlude and Nicol Schraudolph as well as the valuable comments of the anonymous referees.

REFERENCES

[1] F. S. Al-Khaled, "Optimal radio channel assignment through the new binary dynamic simulated annealing algorithm," in *International Journal of Communication Systems*, 11st ed., 1998, pp. 327–336.

[2] E. Amaldi, A. Capone, and F. Malucelli, "Discrete models and algorithms for the capacitated location problem arising in umts network planning," in *Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*, ACM. New York: ACM Press, 2001, pp. 1–8.

[3] D. Beckmann and U. Killat, "Frequency planning with respect to interference minimization in cellular radio networks," TD(99)032, Vienna, Tech. Rep. COST259, 1999.

[4] ——, "A new strategy for the application of genetic algorithms to the channel-assignment problem," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 4, pp. 1261–1269, 1999.

[5] L. Davis, Ed., *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.

[6] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *Parallel Problem Solving from Nature – PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin: Springer, 2000, pp. 849–858.

[7] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, and E. Toto, "Frequency assignment in mobile radio systems using branch-and-cut techniques," *European Journal of Operational Research*, vol. 123, pp. 241–255, 2000.

[8] L. C. P. Floriani and G. R. Mateus, "An optimization model for the bst location problem in outdoor cellular and pcs systems," in *Proceedings of the 15th International Teletraffi c Congress ITC*, 1997.

[9] C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, S. Forrest, Ed. San Mateo, CA: Morgan Kaufmann, 1993, pp. 416–423.

[10] M. Galota, C. Glasser, S. Reith, and H. Vollmer, "A polynomial-time approximation scheme for base station positioning in UMTS networks," in *Proceedings 5th Discrete Algorithms and Methods for Mobile Computing and Communications*, 2000, pp. 52–59.

[11] C. Glasser, S. Reith, and H. Vollmer, "The complexity of base station positioning in cellular networks," in *ICALP Satellite Workshops*. Geneva, Switzerland: Carleton Scientifi c, 2000, pp. 167–178.

[12] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched pareto algorithm for multiobjective optimization," in *Proc. of the First IEEE Conf. on Evolutionary Computation*, vol. 1. Piscataway, NJ: IEEE Service Center, 1994, pp. 82–87.

[13] B. Jaumard, O. Marcotte, and C. Meyer, "Mathematical models and exact methods for channel assignment in cellular networks," in *Telecommunications Network Planning, chapter 13*, B. Sansáo and P. Soriano, Eds. Boston: Kluver Academic Publishers, 1999, pp. 239–255.

[14] J. Kalvenes and R. Gupta, "Hierarchical cellular network design with channel allocation," 2000, to appear: Informs Select Proceedings of the 2000 Telecommunications Conference.

[15] K. Mehlhorn and S. Näher, *The LEDA Platform of Combinatorial and Geometric Computing*. Cambridge: Cambridge University Press, 1999.

[16] V. Pareto, *Cours D'Economie Politique*. Lausanne: F. Rouge, 1896.

[17] B. J. Ritzel, J. W. Eheart, and S. Ranjithan, "Using genetic algorithms to solve a multiple objective groundwater pollution containment problem," *Water Resources Research*, vol. 30, no. 5, pp. 1589–1603, 1994.

[18] P. Ross, D. Corne, and H.-L. Fang, "Improving evolutionary timetabling with delta evaluation and directed mutation," in *Parallel Problem Solving from Nature – PPSN III*, Y. Davidor, H.-P. Schwefel, and R. Männer, Eds. Berlin: Springer, 1994, pp. 556–565, Lecture Notes in Computer Science 866.

[19] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. of the First Int. Conf. on Genetic Algorithms*, J. J. Grefenstette, Ed. Hillsdale, NJ: Lawrence Erlbaum, 1985, pp. 93–100.

[20] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.

[21] G. Syswerda and J. Palmucci, "The application of genetic algorithms to resource scheduling," in *Proc. of the Fourth Int. Conf. on Genetic Algorithms*, R. K. Belew and L. B. Booker, Eds. San Mateo, CA: Morgan Kaufmann, 1991, pp. 502–508.

[22] G. Szabo, N. Weicker, and P. Widmayer, "Base station transmitter placement with frequency assignment: an evolutionary approach (extended abstract)," in *Advances in Nature-Inspired Computation: The PPSN VII Workshops*, D. Corne, G. Fogel, W. Hart, J. Knowles, N. Krasnogor, R. Roy, J. Smith, and A. Tiwari, Eds. Reading, UK: PEDAL (Parallel, Emergent & Distributed Architectures Lab), University of Reading, 2002, pp. 47–48.

[23] K. Tutschku, T. Leskien, and P. Tran-Gia, "Traffi c estimation and characterization for the design of mobile communication networks," University of Würzburg Institute of Computer Science Research Report Series, Würzburg, Tech. Rep. 171, 1997.

[24] K. Tutschku, R. Mathar, and T. Niessen, "Interference minimization in wireless communciation systems by optimal cell site selection," in *Proceedings EPMCC'99*, Paris, 1999, pp. 208–213.

[25] D. A. Van Veldhuizen and G. B. Lamont, "Multiobjective optimization with messy genetic algorithms," in *Proc. of the 2000 ACM Symposium on Applied Computing*. New York: ACM, 2000, pp. 470–476.

[26] N. Weicker, *Qualitative No Free Lunch Aussagen für Evolutionäre Algorithmen*. Göttingen: Cuvillier, 2001.

[27] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Trans. on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[28] T. Yu and P. Bentley, "Methods to evolve legal phenotypes," in *Parallel Problem Solving from Nature – PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin: Springer, 1998, pp. 280–291, Lecture Notes in Computer Science 1498.

[29] X. Zhou and T. Nishizeki, "Effi cient algorithms for weighted colorings of series-parallel graphs," in *Algorithms and Computation, 12th International Symposium, ISAAC 2001*, P. Eades and T. Takaoka, Eds. Berlin: Springer, 2001, pp. 514–524.

[30] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control*, K. Giannakoglou, D. Tsahalis, J. Periaux, K. Papailiou, and T. Fogarty, Eds. Barcelona, Spain: CIMNE, 2002, p. ?

[31] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

Figure captions

- Fig. 1. Example for co-channel and adjacent-channel interference
- Fig. 2. Recombination
- Fig. 3. Replacement strategy for case 1.
- Fig. 4. Replacement strategy for case 2.
- Fig. 5. Replacement strategy for case 3.
- Fig. 6. Demand node distribution in city Zürich
- Fig. 7. NSGA-II: Experiments using random mutations only.
- Fig. 8. SPEA2: Experiments using random mutations only.
- Fig. 9. stEAPT: Experiments using random mutations only.
- Fig. 10. NSGA-II: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.
- Fig. 11. SPEA2: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.
- Fig. 12. stEAPT: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.
- Fig. 13. NSGA-II: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.
- Fig. 14. SPEA2: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.
- Fig. 15. stEAPT: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.

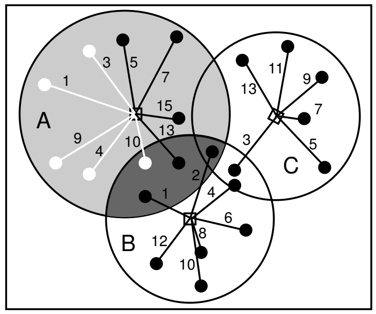

Fig. 1. Example for co-channel and adjacent-channel interference

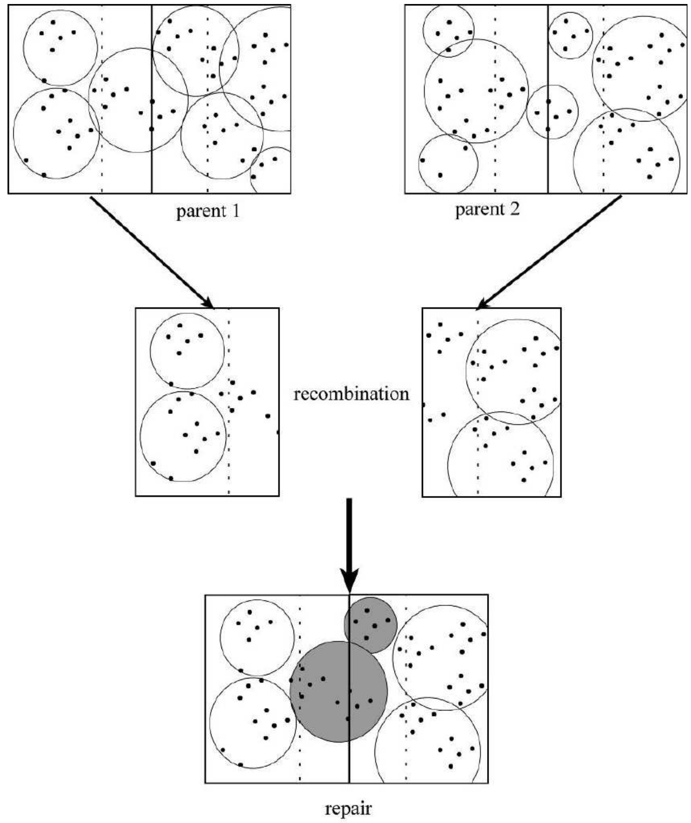

Fig. 2. Recombination

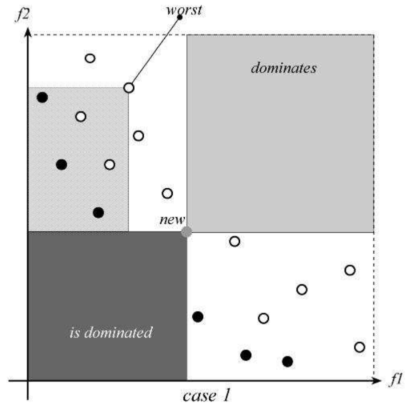

Fig. 3. Replacement strategy for case 1.

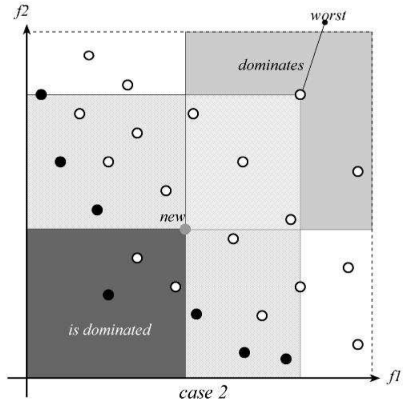

Fig. 4. Replacement strategy for case 2.

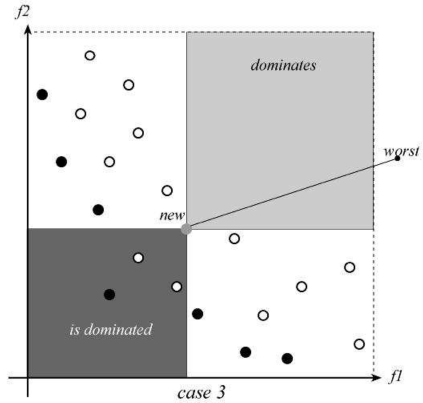

Fig. 5. Replacement strategy for case 3.

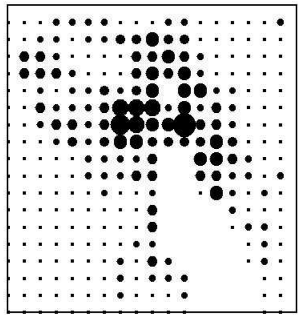

Fig. 6. Demand node distribution in city Zürich

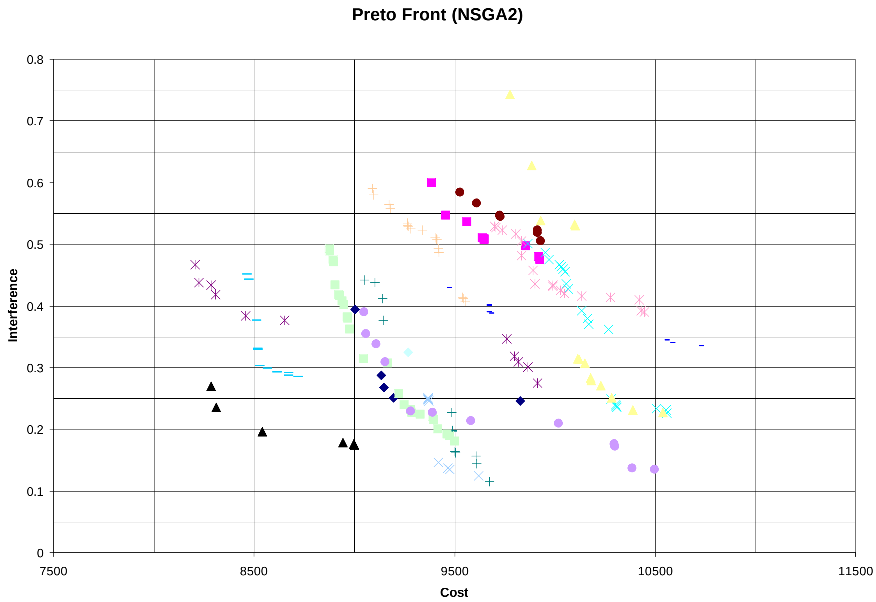

Fig. 7. NSGA-II: Experiments using random mutations only.

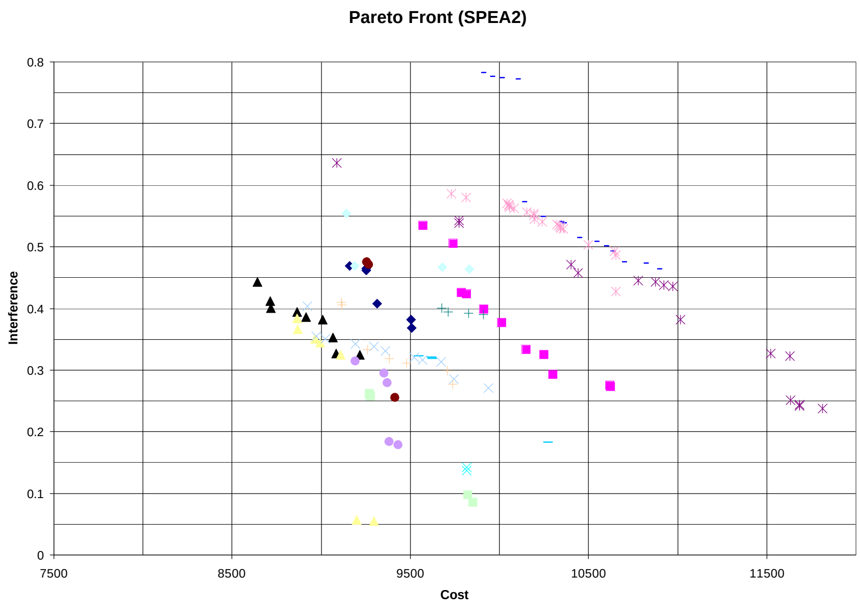

Fig. 8. SPEA2: Experiments using random mutations only.

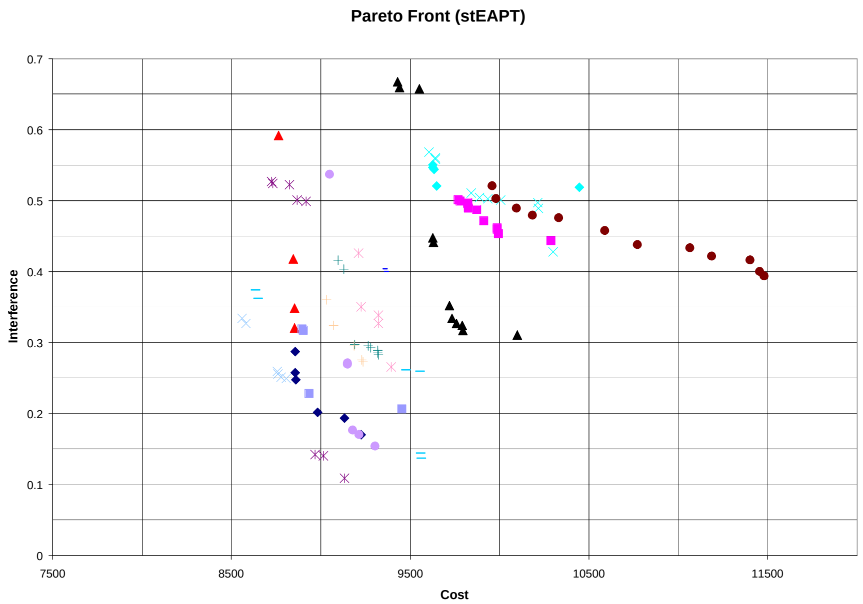

Fig. 9. stEAPT: Experiments using random mutations only.

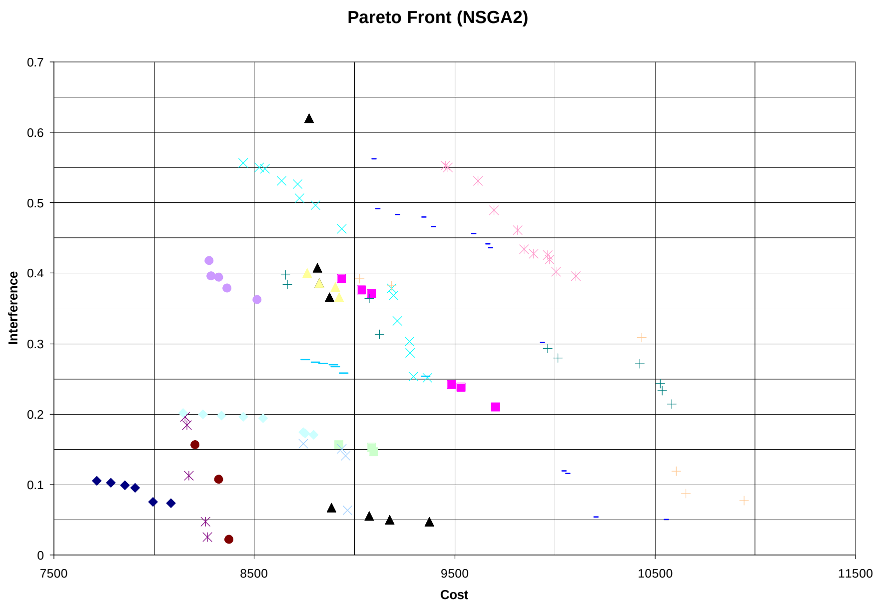

Fig. 10. NSGA-II: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.

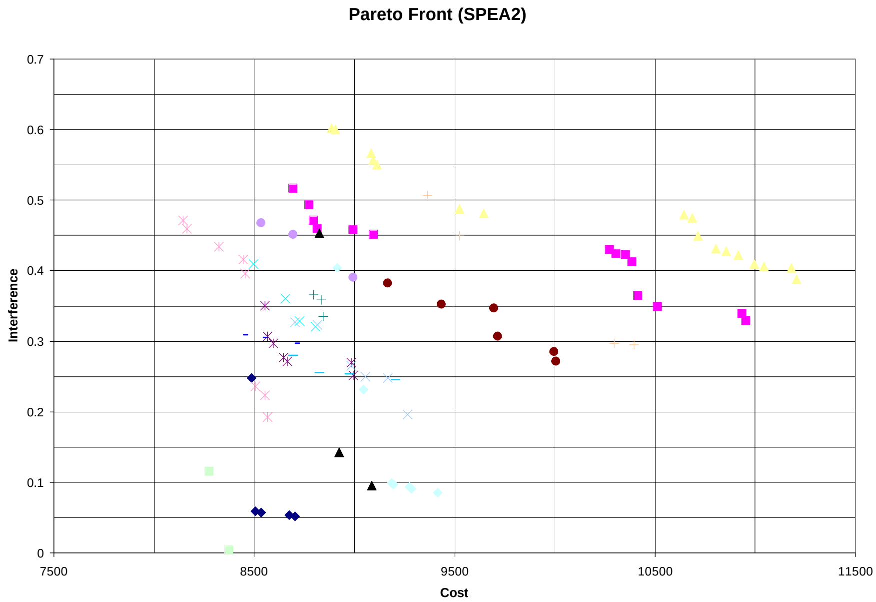

Fig. 11. SPEA2: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.

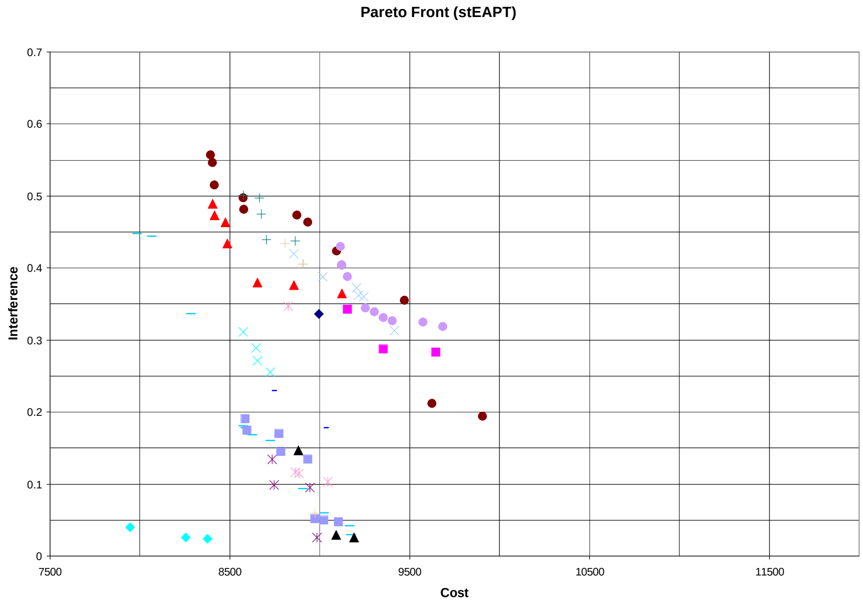

Fig. 12. stEAPT: Experiments using a combination of 50% random mutations, 50% directed mutations and no recombination.

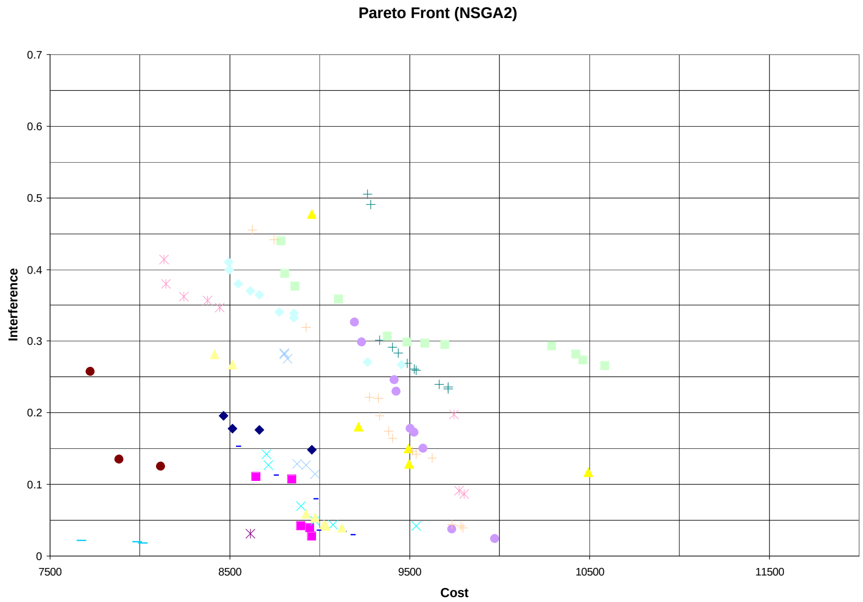

Fig. 13. NSGA-II: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.

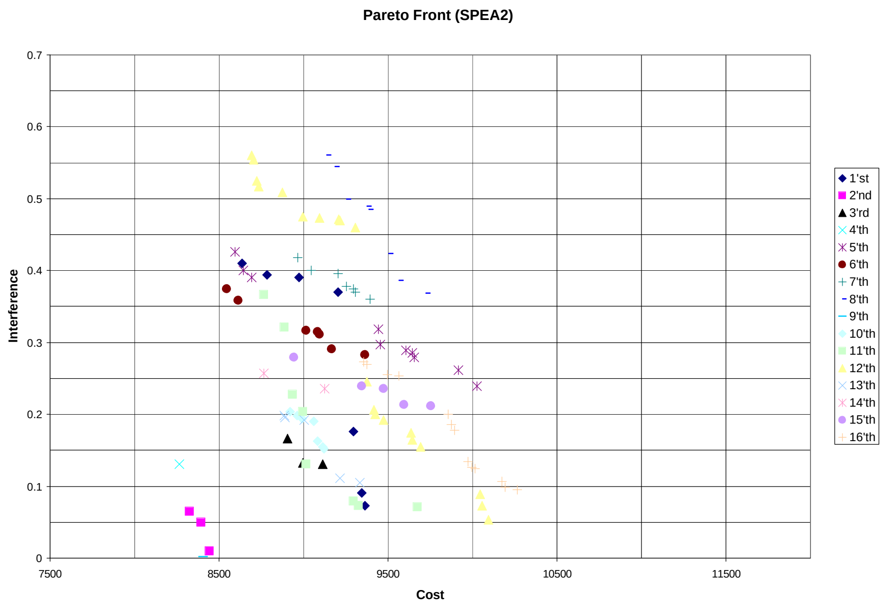

Fig. 14. SPEA2: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.

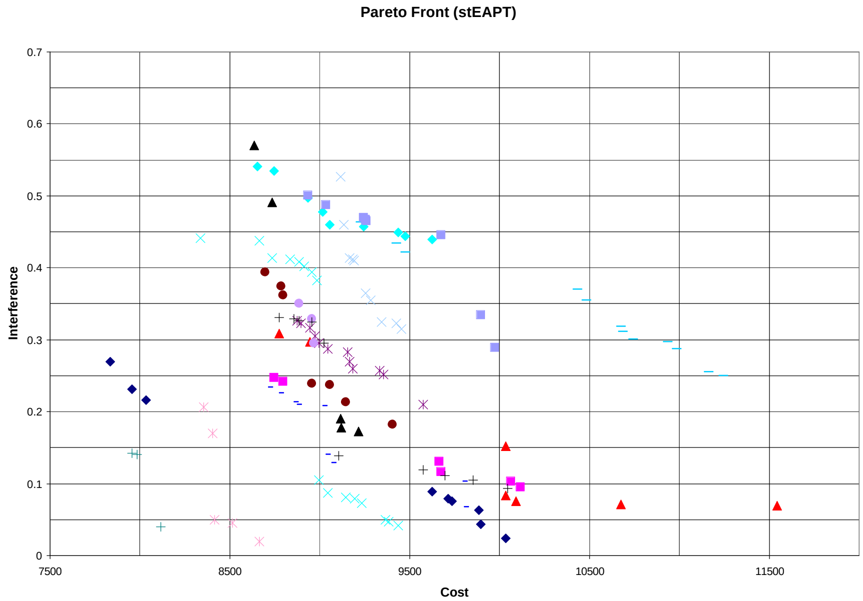

Fig. 15. stEAPT: Experiments using a combination of 30% random mutations, 30% directed mutations and 40% recombination.

Author Biographies

Nicole Weicker

Nicole Weicker received her M.Sc. degree in mathematics from the University of Oldenburg, Germany, in 1993 and her PhD degree in computer science from the University of Stuttgart, Germany, 2001.

She is currently a research fellow at the University of Stuttgart. Her research interests include evolutionary computation and didactics of informatics.

Gabor Szabo

Gabor Szabo received the B.Sc. degree in computer science from the Technical University of Cluj-Napoca, Romania, in 2000.

He is currently a research and teaching assistant at the Swiss Federal Institute of Technology (ETH) at the Institute for Theoretical Computer Science, working on his doctoral thesis under the supervision of Prof. Peter Widmayer.

Karsten Weicker

Karsten Weicker received his M.Sc. degrees in computer science from the University of Massachusetts and the University of Stuttgart, Germany, 1997.

He is currently a research assistant at the University of Stuttgart. His research interests are evolutionary computation and dynamic optimization. He has published more than 15 articles in this area and is also the author of a German text book on evolutionary computation.

Peter Widmayer

Peter Widmayer got his Ph.D. from technical U. Karlsruhe, Germany, in 1983. He spent a postdoc year at IBM T.J. Watson Research Center. Then, he worked at U. Karlsruhe where he finished his habilitation in 1986. From 1988 until 1992, he was a professor of Computer Science at U. Freiburg, Germany, and since then he is with ETH Zurich, Switzerland. He is interested in algorithms and data structures for combinatorial and geometric problems that arise from practical applications.