Fast Approximation Methods for Sales Force Deployment

Andreas Drexl • Knut Haase
Institut für Betriebswirtschaftslehre, Christian-Albrechts-Universität zu Kiel, Olshausenstraße 40, 24118 Kiel, Germany
drexl@bwl.uni-kiel.de • haase@bwl.uni-kiel.de

Sales force deployment involves the simultaneous resolution of four interrelated subproblems: sales force sizing, salesman location, sales territory alignment, and sales resource allocation. The first subproblem deals with selecting the appropriate number of salesman. The salesman location aspect of the problem involves determining the location of each salesman in one sales coverage unit. Sales territory alignment may be viewed as the problem of grouping sales coverage units into larger geographic clusters called sales territories. Sales resource allocation refers to the problem of allocating scarce salesman time to the aligned sales coverage units. All four subproblems have to be resolved in order to maximize profit of the selling organization. In this paper a novel nonlinear mixed-integer programming model is formulated which covers all four subproblems simultaneously. For the solution of the model we present approximation methods capable of solving large-scale, real-world instances. The methods, which provide lower bounds for the optimal objective function value, are benchmarked against upper bounds. On average the solution gap, i.e., the difference between upper and lower bounds, is about 3%. Furthermore, we show how the methods can be used to analyze various problem settings of practical relevance. Finally, an application in the beverage industry is presented.
(*Marketing Models*; *Sales Force Sizing*; *Salesman Location*; *Sales Territory Alignment*; *Sales Resource Allocation*; *Application/Distribution of Beverages*)

1. Introduction

In many selling organizations, sales force deployment is a key instrument in allowing sales management to improve profit. In general, sales force deployment is complicated and has attracted much analytical study. It involves the concurrent resolution of four interrelated subproblems: Sizing the sales force calls for selecting the appropriate number of salesmen. The salesman location aspect of the problem involves determining the location of each salesman in one of the available sales coverage units (SCUs). Sales territory alignment may be viewed as the problem of grouping SCUs into larger geographic clusters called sales territories. Sales resource allocation refers to the problem of allocating salesman time to the assigned SCUs. Research has yielded several models and methods that can be helpful to sales managers.

This paper tackles all four subproblems simultaneously to maximize the profits of the selling organization. To do so, we develop a novel nonlinear mixed-integer programming model. For the solution of the model we present approximation methods capable of solving large-scale, real-world instances.

The paper is structured as follows: In §2 we review previous research. In §3 the problem setting is described in terms of a nonlinear mixed-integer programming model. A fast method for approximately solving large-scale problem instances is presented in §4. The results of an in-depth experimental study are covered in §5. Section 6 discusses insights for market-

1526-5501/99/4510/1307$05.00

ing management. In §7 we describe the results of an application to a practical case arising in the beverage industry. A summary and conclusions are given in §8. In Appendix A we provide a mixed-integer linear programming model for computing upper bounds. In Appendix B the procedures are described formally. Appendix C covers special cases and extensions.

2. Background and Previous Research

The selection of the SCUs depends upon the specific application. SCUs are usually defined in terms of a sales force planning unit for which the required data can be obtained. Counties, zip codes, and company trading areas are some examples of SCUs (cp., e.g., Zoltners and Sinha 1983, Churchill et al. 1993). Note that it is beneficial to work with aggregated sales response functions on the level of SCUs rather than with individual accounts, because substantially fewer response functions need to be estimated and the model size does not explode (cp., e.g., Skiera and Albers 1996).

Sales resource allocation models consist of several basic components, i.e., sales resources, sales entities, and sales response functions. Specific definitions for these components yield specific sales resource allocation models; cp. Zoltners and Sinha (1980) and Albers (1989).

The sales force sizing subproblem has been addressed, for example, by Lodish (1980). However, the salesman location subproblem has not been the subject of research in the context of sales force deployment so far; our work closes this gap. Among the four interrelated subproblems, the alignment subproblem has attracted the most attention so far. Several approaches concerning this problem have been published. They can be divided into those that depend upon heuristics and those that utilize a mathematical programming model. Heuristics have been proposed by Heschel (1977), among others. Two types of mathematical programming approaches have been developed. Shanker et al. (1975) formulated a set-partitioning model. Alternatively, the models proposed by Segal and Weinberger (1977) as well as those of Zoltners and Sinha (1983) are SCU-assignment models. The sales resource allocation subproblem has been analyzed by Beswick (1977) and Zoltners et al. (1979).

Some of the papers published so far on the alignment subproblem have been aimed at aligning sales territories in an attempt to balance one or several attributes. The most popular balancing attributes are sales potential or the workload of the salesmen. A detailed discussion of the shortcomings of the balancing approaches can be found in Skiera and Albers (1996, 1998).

Glaze and Weinberg (1979) address the three subproblems of locating the salesmen, aligning accounts, and allocating calling time. More specifically, they present the procedure TAPS, which seeks to maximize sales for a given salesforce size while also attempting to achieve equal workload between salespersons and, in addition, minimize total travel time.

Recently, Skiera and Albers (1994, 1996) formulated a model that addresses the sales territory alignment and the sales resource allocation problem simultaneously. For the solution of their model they propose a simulated annealing heuristic. The objective of their model is to align SCUs and to allocate resources in such a way that sales are maximized. Yet, their model only addresses two of the subproblems covered by our model; hence, it is a special case. While both approaches allow the computation of feasible solutions, i.e. lower bounds, an additional advantage of our approach is that it also allows the determination of upper bounds. In our opinion, it is theoretically more sound to compare lower with upper bounds than to compare lower bounds with each other.

3. Nonlinear Mixed-Integer Programming Model

The greater the size of the sales force, the more customers can be visited, which in turn has a positive impact on sales. On the other hand, increasing the size of sales force tends to increase the operational costs per period. In addition, the number of possible calls to customers, the operational costs, and the salesmen's resource (time) that might be allocated to customers is affected by the location of the salesmen, too. To make things even more complicated, the alignment decision is very important for all these issues as well. Clearly, we have to take care of all the mutual interactions of the different factors affecting the quality of the overall sales force deployment. The aim of the following is to provide a formal model which relates all the issues to each other.

Let us assume that the overall sales territory has already been partitioned into a set of J SCUs. The SCUs have to be grouped into mutually disjointed sales territories (clusters), such that each SCU $j \in J$ is assigned to exactly one cluster and the SCUs of each cluster are connected. In each cluster a salesman has to be located in one of the assigned SCUs, called sales territory center. Note that *connected* means that we can "walk" from a location to each assigned SCU without crossing another sales territory. $I \subseteq J$ denotes the subset of SCUs that are potential sales territory centers. To simplify notation, $i \in I$ denotes both the sales territory center i and the salesman located in SCU i.

In practice, selling time consists of both calling time and travel time. For notational purposes, let $z_{i,j}$ denote the calling time per period that is spent by salesman i to visit customers in SCU j. Further, assume $b_j \in [0, 1]$ to denote the calling time elasticity of SCU j and $g_j > 0$ a scaling parameter. Then

$$S_{i,j} = g_j(z_{i,j})^{b_j} \tag{1}$$

defines expected sales $S_{i,j}$, $i \in I$, $j \in J$, as a function of the time to visit customers. More precisely, Equation (1) relates $z_{i,j}$ to $S_{i,j}$ for all sales territories $i \in I$ and SCUs $j \in J$. Hence, via b_j it is possible to take care of the fact that a firm's competitive edge might be different in different SCUs. Note that expected sales are defined via concave rather than s-shaped functions, as is assumed to be the case with individual accounts (cp. Mantrala et al. 1992).

Let $t_{i,j}$ denote the selling time of salesman $i \in I$ in SCU $j \in J$. Note that $t_{i,j}$ includes the time to travel from SCU i to SCU j, the time to travel to customers in SCU j, and the customer calling time, respectively. Then, $p_{i,j} = z_{i,j}/t_{i,j}$ relates the calling time $z_{i,j}$ to the selling time $t_{i,j}$. Substituting $z_{i,j}$ in Equation (1) for $p_{i,j}t_{i,j}$ yields

$$S_{i,j} = g_j(p_{i,j}t_{i,j})^{b_j} = c_{i,j}(t_{i,j})^{b_j}. \tag{2}$$

Note that Equation (2) was first proposed by Skiera and Albers (1994). In Equation (2), the parameter

$$c_{i,j} = g_j(p_{i,j})^{b_j} \tag{3}$$

is introduced. The symbol $c_{i,j}$ measures the sales contribution when SCU j is part of sales territory i where $c_{i,j}$ is a function of $p_{i,j}$. This is best illustrated as follows: Suppose that for salesman i the travel times to customers in SCUs j and k are different. Then, in general, $p_{i,j}$ and $p_{i,k}$ will also be different. Clearly, this produces different parameters $c_{i,j}$ and $c_{i,k}$—and puts emphasis on the location decision.

Equations (2) and (3) introduce the idea that selling time is a constant proportion of the travel time within an SCU, and this proportion is independent of the amount of effort allocated to an SCU. Conditions under which this is likely to be met are a relatively homogeneous customer base within an SCU, not too many nontraversible barriers within an SCU, etc.

Now we are ready to state the model formally. We summarize the model parameters

J: set of SCUs, indexed by j;

I: set of SCUs ($I \subseteq J$) for locating salesmen, indexed by i;

$\mathcal{N}_j$: set of SCUs which are adjacent to SCU j;

f_i: per period fixed cost for locating a salesman in SCU i;

$c_{i,j}$: expected contribution if SCU j is covered by the salesman located in i;

b_j: calling time elasticity of SCU j;

T_i: total selling time available per period for salesman i;

introduce the decision variables

$x_{i,j}$: $=1$, if SCU j is assigned to the salesman located in SCU i ($x_{i,j} = 0$, otherwise);

$t_{i,j}$: selling time allocated by the salesman located in SCU i to SCU j ($t_{i,j} \geq 0$);

and formulate a nonlinear mixed-integer programming (NLP) model as follows:

$$\text{maximize } Z_{\text{NLP}}(x, t) = \sum_{i \in I} \sum_{j \in J} c_{i,j}(t_{i,j})^{b_j} - \sum_{i \in I} f_i x_{i,i} \tag{4}$$

subject to

$$t_{i,j} \leq T_i x_{i,j} \quad (i \in I, j \in J), \tag{5}$$

$$\sum_{j \in J} t_{i,j} \leq T_i x_{i,i} \quad (i \in I), \tag{6}$$

$$\sum_{i \in I} x_{i,j} = 1 \quad (j \in J), \tag{7}$$

$$\sum_{j \in \cup_{v \in V} \mathcal{N}_v - V} x_{i,j} - \sum_{k \in V} x_{i,k} \geq 1 - |V| \tag{8}$$

$$(i \in I,\ V \subseteq J - \mathcal{N}_i - i),$$

$$x_{i,j} \in \{0, 1\} \quad (i \in I,\ j \in J), \tag{9}$$

$$t_{i,j} \geq 0 \quad (i \in I,\ j \in J). \tag{10}$$

Objective (4) maximizes sales while taking the fixed cost of the salesman locations into account—and hence maximizes profit contribution or profit for short. The salesman i is allowed to allocate selling time to SCU j only when SCU j is assigned to him (cp. Equation (5)). Equation (6) guarantees that the maximum workload per period (consisting of travel and call time) and per salesman is regarded. Equation (7) assigns each SCU to exactly one of the salesmen. Equation (8) guarantees that all the SCUs assigned to one sales territory are connected to each other. Equations (9) and (10) define the decision variables appropriately.

The model (4) to (10) has linear constraints, but a nonlinear objective. Furthermore, we have continuous and binary decision variables. Therefore, there is no chance of solving this model with standard solvers. In Appendix A it is shown how the objective function can be linearized in order to make the model accessible to mixed-integer programming (MIP) solvers. This makes it possible to compute upper bounds for medium-sized problem instances that in turn facilitates the evaluation of the performance of the heuristics.

As already mentioned, Equation (8) guarantees that all the SCUs assigned to one sales territory are connected with each other. These equations work similarly to constraints destroying short cycles in traveling salesman model formulations. Unfortunately, their number is exponential in the worst case; hence, we proceed as follows: Start with all constraints except (8). Solve the model and check the solution as to whether some of the constraints (8) are violated. Add them to the constraint set, reoptimize, and so on until no constraint is violated. Consider, for instance, the following situation where SCUs 2 to 4 are adjacent to SCU 5, but SCU 1 is not. Now assume that SCUs 1 and 5 are assigned to salesman (SCU) 5 in the current solution, for example. Then the constraint $x_{5,2} + x_{5,3} + x_{5,4} - x_{5,1} \geq 0$ enforces (for $V = \{1\}$) that at least one of the neighbors 2, 3, or 4 of SCU 5 has to be assigned to salesman 5. Adding this constraint to the preexisting constraint set then probably generates a feasible solution via reoptimization. In general, only a couple of iterations are necessary in order to come up with a solution where all adjacency constraints (8) are fulfilled. Clearly, it would be sufficient to take care of connected subsets $V \subseteq J - \mathcal{N}_i - i$, $i \in I$, of SCUs only. Obviously, this procedure is only practical when applied to the MIP-approximation presented in Appendix A. When solving NLP with approximation methods Equation (8) is satisfied "directly."

Adjacency of SCUs "physically" has the following meaning. Without loss of generality assume that the shape of the SCUs is described by polygon lines. Then, two SCUs are adjacent if they share at least one line of the polygon. Practically, this implies that two such SCUs can be assigned to the same salesman. In §5.1, where we describe the instance generator used in this study, "adjacency" is formally defined. Moreover, in §7 we provide insight into what adjacency means in practice with respect to nontraversible obstacles, for example.

So far we have not mentioned the following assumption covered by our model: $x_{i,i} = 1$ means that SCU i is assigned to the salesman located in sales territory i. In other words, $x_{i,i} = 1$ does not only tell us where to locate salesman i, it also defines how to align SCU i. This assumption is justified with respect to practice. Moreover, we assume by definition of the binary alignment variables $x_{i,j} \in \{0, 1\}$ that accounts are exclusively assigned to individual salesmen. Note that this is an assumption in marketing science and marketing management for several appealing reasons.

Clearly, all the parameters of the sales response function (1) have to be estimated. This can be done as follows if a sales territory alignment has already existed for several periods, i.e., if our concern is to rearrange an already existing sales territory alignment. Then data for each SCU about the sales, the time to travel to customers, as well as the time to visit the customers are already available. Usually, this data can be extracted from sales reports. In this situation b_j and g_j can be estimated as follows. Transform Equation (1) to Equation (11):

$$\ln(S_{i,j}) = \ln(g_j) + b_j \ln(z_{i,j}), \tag{11}$$

and then calculate estimates of b_j and g_j via linear regression. Finally, for the computation of $c_{i,j}$ we need estimates of $p_{i,j}$. In this regard, the time to travel from SCU i to SCU j and the time to visit the customers within SCU j are required. If salesman i has already covered SCU j in the past we just have to look at his sales reports. Otherwise, we assume that the time to travel within an SCU is independent of the salesman. Then the only data required for a salesman $k \neq i$ is the time to travel from k to i. This is easily available, e.g., from commercial databases or simply by assuming that the travel time is proportional to the travel distance. In the case of the sales territory having to be designed from scratch, more effort is necessary. Unfortunately, going into detail is beyond the scope of this paper.

Our model addresses a medium-term problem. On a short-term basis, changes of specific salespersons probably occur frequently. In this situation, marketing management in general will not resize, relocate, realign, or reallocate the overall sales territory after each change. Hence, sales force deployment has to be done with respect to salespersons whose effectiveness is "average." Having deployed the sales force by means of our medium-term model, then operationally some further issues need to be addressed by each salesman periodically. He has, for instance, to decide upon the routing and scheduling when visiting his customers. Rush-hours, etc. must be taken into account. Also, the total selling time may vary from period to period, having the agency theorists view this to be a salesmans choice variable. Clearly, the outcome of such operational decisions will, in some cases, question the input parameters of the medium-term model. Nevertheless, the deployment of the sales force while taking average data into account is justified. Note that in §6 we will show that the results are very robust with respect to incorrect estimates of parameters (cp. Table 3).

The four interrelated subproblems are addressed in our model by the decision variables $x_{i,j}$ and $t_{i,j}$. Let $x^*_{i,j}$ and $t^*_{i,j}$ denote an optimal solution for a given problem instance. Then we can make the following statements: (i) The optimal size of the sales force $|\mathcal{I}|$ which corresponds to the optimal number of sales territories (clusters) is given by the cardinality of the set $\mathcal{I} = \{i | x^*_{i,i} = 1\}$. (ii) For each of the sales territories in the set $\mathcal{I}$ the SCU i with $x^*_{i,i} = 1$ is the optimal location of the salesman, i.e. the optimal sales territory center. (iii) For each sales territory $i \in \mathcal{I}$ the optimal set $\mathcal{J}_i$ of aligned sales territories or SCUs is given by $\mathcal{J}_i = \{j | x^*_{i,j} = 1\}$. (iv) Finally, $t^*_{i,j} > 0$ is the optimal sales resource allocation for $i \in \mathcal{I}$ and $j \in \mathcal{J}_i$. This interpretation of an optimal solution $x^*_{i,j}$ and $t^*_{i,j}$ illustrates that the model is "lean" in the sense that two types of decision variables cover all the four subproblems of interest. This suggests that the model is in fact a suitable representation of the overall decision problem. Moreover, it comprises the first step towards a solution of the problem.

The aim of the following section is to present heuristic methods which balance computational tractability with optimality.

4. Approximation Methods

This section discusses a solution approach which has been developed specifically for the model. Two reasons led to this development. First, standard methods of mixed-integer programming seem to lend themselves to solving the linearized version of the model. However, even for modestly sized problems the formulation translates into very large mixed-integer programs which in turn result in prohibitive running times. In fact it is conjectured that, except for smaller problem sizes, no exact algorithm will generally produce optimal solutions in a reasonable amount of time. Second, apart from exact methods, no heuristic is available for solving the model so far. The simulated annealing procedure of Skiera and Albers (1994, 1996) solves two of our subproblems, i.e. the sales territory alignment and the sales resource allocation. Unfortunately, it does not tackle the sales force sizing and the salesman location subproblems. In addition, although dealing only with two of the four subproblems, in general the running times of the simulated annealing procedure do not allow the solution of large-scale problem instances in a reasonable amount of time.

The basic idea of our heuristic is very simple and can be summarized as follows: For a given sales force size, start with an initial location for each salesman. Try alternate sales locations to obtain improved locations. For each set of salesman locations align the sales territory in a greedy manner with an SCU going to the salesman based on the resulting marginal increase in the objective function. This procedure is repeated for different sales force sizes (exploiting the fact that the objective function is concave). Once an initial feasible solution is obtained, it is improved by testing feasible exchanges for territory alignment. The heuristic may be characterized as a construction and improvement approach. It consists of the Procedure Construct and the Procedure Improve. The Procedure construct determines the sales force size and hence the number of salesmen. In addition, it calls two other procedures: The Procedure Locate which computes the SCU in which each salesman has to be located and the Procedure Align which aligns the SCUs to the already existing sales territory centers. The Procedure Improve systematically interchanges adjacent SCUs of two different clusters. This way it improves the feasible solution, which is the outcome of the Procedure Construct.

Note that the sales resource allocation subproblem can be solved by Equation (13) or by Equation (14) as soon as all sales territories are aligned. Now, we describe the procedures designed to generate feasible solutions followed by the description of Equations (13) and (14). Then the improvement procedure will be presented.

4.1. Compute Feasible Solution

Recall that J denotes the set of SCUs, $I = \{i_1, \ldots, i_{|I|}\}$ is the set of SCUs that are potential locations, and $\mathcal{N}_j$ denotes the set of SCUs that are adjacent to SCU j, respectively. In addition, let denote

$\underline{S}$: the minimum number of sales territory centers that might be established ($\underline{S} \geq 0$);

$\bar{S}$: the maximum number of sales territory centers that might be established ($\bar{S} \leq |I|$);

s: the "current" number of sales territory centers ($\underline{S} \leq s \leq \bar{S}$);

I_1: the set of selected locations ($|I_1| = s$, $I_1 = \{i_1, \ldots, i_s\}$);

I_0: the set of nonselected locations ($I_0 \cap I_1 = \emptyset$, $I_0 \cup I_1 = I$);

$L(I_1)$: the locations (i.e. SCUs) of the sales territory centers $i \in I_1$;

$j(i)$: the SCU j where sales territory center $i \in I_1$ is located;

$i(j)$: sales territory center i to which SCU j is assigned;

J_0: the set of SCUs that are not yet aligned (initially $J_0 = J \backslash \cup_{i \in I_1} j(i)$);

J_i: the set of SCUs that are aligned to sales territory center $i \in I_1$;

C_i: sum of sales contributions of location $i \in I_1$ ($C_i = \sum_{j \in J_i} c_{i,j}$);

LB: a lower bound on the optimal objective function value.

Based on these definitions, the set $\mathcal{A}_i$ of SCUs, which might be aligned to sales territory center i, may be formalized according to Equation (12):

$$\mathcal{A}_i = \cup_{k \in J_i} \mathcal{N}_k \cap J_0. \tag{12}$$

Note that the number of sales territory centers equals the number of salesmen (i.e. the sales force size), which in turn equals the number of locations. Therefore, some of the newly introduced parameters are superfluous, but this redundancy will be helpful for the description of the procedures.

In the following, Z will denote the objective function value of a feasible solution at hand. Clearly, Z is a function of the decision variables $x_{i,j}$ and $t_{i,j}$. The algorithms do not operate on the set of $x_{i,j}$ variables, only the $t_{i,j}$ variables will be used directly. Subsequently, it will be more convenient to express the $x_{i,j}$ decisions also partly in terms of the number of salesman s and in terms of $L(I_1)$, respectively. Redundancy will simplify the formal description and ease understanding substantially. With respect to this redundancy, $Z(\ldots)$ will be used in different variants, but from the local context it will be evident what it stands for.

We introduce a global variable lose[h, i], $h \in I$, $i \in I$, which is used for locating the salesmen in the set of potential locations. More precisely, this variable counts the number of times the best current nonselected "loser" location does not succeed in pushing away the selected "winner" location. That is, alternate locations are selected based on the function lose[h, i] for obtaining improved locations. The current "best" loser i is defined as the nonselected location $i \in I_0$, which so far has lost the least number of times against the winner $i_k \in I_1$. Loser i is supposed to have the best chances of pushing i_k away. In other words, i acts as a "challenger" to i_k. If i does not succeed in pushing i_k away then the global variable is incremented and, hence, the chance of i being reselected as a challenger decreases. The global variable lose[h, i] is used in the Procedure Locate as in tournament selection. The tournament is finished when the best nonselected "player" does not win against any other "player" $i_k \in I_1$. This is an effective means of selecting some elements of a probably large set quickly.

A formal description of the Procedure Construct is given in Table 6 (cp. Appendix B). It consists of an overall loop which updates the current number s of salesmen under consideration. Then it passes calls to Procedure Locate and to Procedure Align and afterwards evaluates the resource allocation by Equation (13) or (14). Finally, the objective function values $Z(s, x, t)$ and $Z(s + 1, x, t)$ are compared with the best-known lower bound LB which is updated whenever possible. Note, the number of salesmen s for which a search is performed is, without loss of generality, restricted to the interval $\underline{S} \leq \bar{S}$. Note that for "reasonable" parameters $c_{i,j}$ and f_i the objective function is concave with respect to the sales force size. Therefore, "gradient search" within the interval $\underline{S} \leq s \leq \bar{S}$ is implemented in the Procedure Construct.

When a call to Procedure Locate (cp. Table 7) is passed, we start with $|I_1| = s$, $I_0 = I \backslash I_1$, which implies $I_1 \cap I_0 = \emptyset$ and $I_1 \cup I_0 = I$ and initializes $L(I_1)$. Note that the Procedure Locate uses only the current number s of locations as a calling parameter. The for-loop tells us that, as starting locations $L(I_1)$, the "first" $|I_1|$ elements of the set I of potential locations are chosen. Recall that $i_k \in I_1$ denotes the winner location that the best loser tries to push away. The set $I_1 = (I_1 \backslash i_k) \cup h$ temporarily changes the role of the winner i_k and of the loser h. If h has no success, this operation has to be reversed. The procedure stops when no further improvement of the set of locations can be found within the for-loop. As an outcome we know the locations $L(I_1)$ of the current number s of salesmen.

Capitalizing on the definitions given above a compact description of the Procedure Align is given in Table 8. Within the while-loop one of the not yet aligned SCUs is chosen and aligned to one of the already existing sales territory centers. The Procedure Align is greedy in the sense that the steepest ascent of the objective function is used as the criterion for the choice of the next SCU to be aligned. More precisely, the choice depends on the ratios $c_{h,i}/C_i$, i.e., the rationale is to take care of the relative weights of the expected sales contributions.

Clearly, as a final step of the overall Procedure Construct, the sales resource allocation subproblem has to be solved. This is done by evaluating Equation (13) or (14).

$$t_{i,j} = \frac{(c_{i,j}x_{i,j})^a}{\sum_{h \in J_i} (c_{i,h})^a} \cdot T_i \quad (i \in I, j \in J). \tag{13}$$

In the case of $b_j = b \; \forall j \in J$ where $a = 1/(1 - b)$, Beckmann and Golob (1972) have shown that Equation (13) provides the optimal resource allocation respecting a given total selling time T_i (cp. Einbu 1981 also). In the general case where $b_h \neq b_j$, $h \in J$, $j \in J$, $h \neq j$, allocation is done by Equation (14). The symbols $a_j = 1/(1 - b_j)$ and $\alpha_i = 1/(1 - \beta_i)$ are used for short, where β_i is the "average" elasticity which has to be calculated by bisection search (for details, cp. Skiera and Albers 1994).

$$t_{i,j} = \left(\frac{(c_{i,j}b_j)^{\alpha_i}}{\sum_{h \in J_i} (c_{i,h}b_h)^{a_h}} \cdot T_i \right)^{(a_j/\alpha_i)} \quad (i \in I, j \in J). \tag{14}$$

4.2. Improve Feasible Solution

In general, feasible solutions at hand can easily be improved by the following simple Procedure Improve. For a compact description of the procedure, we define two boolean parameters:

$$\text{add}(J_i, j) = \begin{cases} \text{TRUE} & \text{if } J_i \cup j \text{ is connected,} \\ \text{FALSE} & \text{otherwise,} \end{cases}$$

$$\text{drop}(J_i, j) = \begin{cases} \text{TRUE} & \text{if } J_i \backslash j \text{ is connected and } j \neq j(i), \\ \text{FALSE} & \text{otherwise.} \end{cases}$$

The function add(J_i, j) defines only those alignments as feasible where we *add* SCU j to the sales territory J_i, such that the newly derived sales territory consists of connected SCUs only. Similarly, the function drop(J_i, j) only permits alignments to be feasible when we *drop* SCU $j \neq j(i)$ from sales territory J_i without running into disconnectedness. In other words, both functions define those moves of an SCU j to/from a sales territory J_i as feasible where the outcome does not violate the connectivity requirement. As a consequence, only those SCUs are suspected move candidates that are located on the border of each of the sales territories. In this respect the functions add(J_i, j) and drop(J_i, j) are complementary. As a consequence, the Procedure Improve might be characterized as an interchange method, too. Clearly, the resource allocation $t_{i,j}$ has to be updated with respect to each move by evaluating Equation (13) or (14).

A formal description of the Procedure Improve is given in Table 9. For the sake of compactness, the calling parameter $J = (J_1, \ldots, J_s)$ denotes the vector of sales territory alignments currently under investigation, and $Z(J)$ the corresponding objective function value, respectively. $Z(J|J_{i(j)}\backslash j, J_i \cup j)$ tells us that the objective function value has to be computed with respect to the current alignment under investigation where SCU j is subtracted from sales territory $J_{i(j)}$, while sales territory J_i is augmented by SCU j. Clearly, the computation of the objective function also requires an update of the resource allocation $t_{i,j}$ via Equation (13) or (14).

Apparently, the Procedure Improve belongs to the variety of local search methods. To keep our explanations as simple as possible, we distract our attention from the resource allocation $t_{i,j}$ for the moment. Then, starting with an incumbent sales territory alignment $x = (x_{i,j})$, we search all its neighbors $\hat{x} \in \mathcal{H}(x)$, where $\mathcal{H}(x)$ equals the set of feasible solutions that are properly defined by the functions add(J_i, j) and drop(J_i, j). $\mathcal{H}(x)$ is called neighborhood of x. Searching over all neighbors $\hat{x} \in \mathcal{H}(x)$ in a steepest ascent manner may be characterized as a "best fit strategy." By contrast, a "first fit strategy" might be less time consuming while presumably producing inferior results.

5. Experimental Evaluation

The outline of this section is as follows: First, we elaborate on the instances used in our computational study. Second, we describe how to compute benchmark solutions to judge the performance of the methods presented in the preceeding section. Third, numerical results will be presented.

Even in current literature, the systematic generation of test instances does not receive much attention. Generally, two possible approaches adopted in literature can be found when having to come up with test instances. First, consider practical cases. Their strength is their high practical relevance while the obvious drawback is the absence of any systematic structure allowing the inference of any general properties. Thus, even if an algorithm performs well in some practice cases, it is not guaranteed that it will continue to do so in other instances as well. Second, consider artificial instances. Since they are generated randomly according to predefined specifications, their advantage lies in the fact that fitting them to certain requirements such as given probability distributions poses no problems. However, they may reflect situations with little or no resemblance to any problem setting of practical interest. Hence, an algorithm performing well on several such artificial instances may or may not perform satisfactorily in practice. Therefore, we decided to devise a combination of both approaches, thereby attempting to keep the strengths of both approaches while avoiding their drawbacks. More precisely, first we evaluated the methods on a set of instances which had been generated at random (cp. §5.1 to §5.3). Second, we applied the systematically evaluated methods to a practical case (cp. §7).

5.1. Generation of Instances

We assumed that only two instance-related factors have a major impact on the performance of the algorithms, viz. the cardinality of the set I of potential sales territory centers and the cardinality of the set J of SCUs, respectively. Both factors relate to the "size" of a problem, hence (I, J) denotes the size of an instance.

When generating instances at random a critical part is the specification of a connected sales territory. To do so, we employ the Procedure Generate, which is able to generate a wide range of potential sales territories while preserving connectivity. The basic idea is to define a set $K = \{1, \ldots, 2 \cdot Q\} \times \{1, \ldots, 2 \cdot Q\}$ with $K \geq J$ of unit squares located on a grid. For every unit square $(\alpha, \beta) \in K$, the set of adjacent unit squares $\mathcal{N}_{(\alpha,\beta)}$ or neighbors is defined as follows:

$$\mathcal{N}_{(\alpha,\beta)} = \{\{(\tilde{\alpha}, \tilde{\beta}) \in K \mid |\tilde{\alpha} - \alpha| \leq 1, |\tilde{\beta} - \beta| \leq 1\} \backslash (\alpha, \beta)\}.$$

The Procedure Generate is formally described in Table 10. The set of sales territory centers I and the set of SCUs J are used as calling parameters. Note that, starting with the "central" unit square $\mathcal{M} = \{(Q, Q)\}$, the set $\mathcal{M}$ is incremented until it equals the set of SCUs J, which have to be generated while preserving the connectivity of the sales territory. Similarly to the Procedure Align, $\mathcal{A}$ denotes the set of those unit squares of the grid which are candidates for the alignment to the already generated sales territory. In a last step, the set of sales territories I is chosen at random.

It is easy to verify that the Procedure Generate is capable of producing a large range of sales territories shaped quite differently. Nevertheless, the question is whether this construction process, which basically relies on unit squares and hence on SCUs of equal size, produces instances which are meaningful to the methods to be evaluated. The answer is "yes" because the grouping, i.e. building of larger units, is just what the Procedure Align does.

A summary of the instances treated in the computational study can be given as follows.[1] (i) The set of SCUs J is given by {50, 100, 250, 500}. (ii) The set of potential sales territory centers I is given by {10, 25, 50}. (iii) The scaling parameter g_j is chosen at random from the interval [10, 210]. (iv) The expected sales $c_{i,j}$ are equal to $g_j(p_{i,j})^b$ where $p_{i,j}$ is computed as follows:

$$p_{i,j} = \max\left\{0,\ 0.4 - \frac{|\alpha_i - \alpha_j| + |\beta_i - \beta_j|}{100}\right\};$$

b was set to 0.3 with respect to empirical findings of Albers and Krafft (1992). (v) The fixed costs f_i of sales territory centers are drawn at random from the interval [750, 1,250]. (vi) The maximum workload T_i per period and salesman is set to 1,300 for all $i \in I$. This is an estimate of the annual average time salesmen in Germany have to work (cp. Skiera and Albers 1994). (vii) The lower bound $\underline{S}$ for the number of sales territory centers is set to 0 while the upper bound $\bar{S}$ equals $|I|$.

The adjacency structures generated by our procedure can be summarized as follows: The instances with $J = 500$ (1,000) SCUs have 6.81 (6.94) neighbors on average. An evaluation of the adjacency structure for 1,144 pharmaceutical SCUs in Germany reveals only 5.541 neighbors on average, while the application presented in §7 shows 5.73 neighbors. Hence, the problems generated have a high level of connectivity.

Note that to calculate the scaling parameter g_j at random as described above might not be the best choice if the data is spatially autocorrelated. While it is certainly not that difficult to generalize the generator such that autocorrelation is also covered we do not follow these lines here for the following reason: The practical case described in §7 has spatially autocorrelated data. Solving the practical case with our procedures is by no means more difficult than solving the artificial instances (details are provided below). Hence, we refrain from introducing more parameters to attain a more "realistic" instance generator.

Clearly, only "reasonable" combinations of J and I are taken into account (details are provided below). In addition, because of the computational effort required to attempt all the sizes, only ten instances for each instance class (J, I) were considered in the experiment.

5.2. Computation of Benchmarks

Unfortunately, it is not possible to solve the NLP-model (4) to (10) by the use of a "standard" solver. Hence, even for small-sized problem instances there is

[1] Available via ftp://www.wiso.uni-kiel/pub/operations-research/salesforce.

Table 1 Comparison of Lower and Upper Bounds

		CPLEX		CONIMP		
$\|J\|$	$\|I\|$	UB	CPU	LB	CPU	GAP
50	10	12,043.39	3.20	11,508.89	$\ll 1$	4.46
	25	12,793.16	14.50	12,217.26	0.90	4.50
	50	13,271.83	49.20	12,736.17	2.60	4.04
100	10	26,119.68	10.30	25,610.99	0.50	1.95
	25	28,332.57	47.10	27,619.47	4.10	2.54
	50	29,583.04	172.60	28,464.76	13.50	3.73
250	50	71,185.04	720.68	69,774.65	142.50	1.99
500	50	133,702.41	3,424.34	130,962.26	626.70	2.05

no "direct" way to obtain benchmarks. In Appendix A, the model (4) to (10) is reformulated as a mixed-integer linear programming (MIP-) model, such that upper bounds of the optimal objective of the NLP-model and, hence, benchmarks can be computed. The MIP-model can be solved directly by the use of one of the commercially available MIP-solvers. This way it is possible to compute upper bounds for problems having up to $J = 500$ SCUs and $I = 50$ sales territories in a reasonable amount of time.

5.3. Computational Results

The algorithms have been coded in C and implemented on a 133 Mhz Pentium machine under the operating system Linux. The parameter K of the Procedure Generate is defined as $K = FAC \times |J|$ where $FAC = 1.5$ has been used. Note that $FAC > 1$ serves to generate sales territories where not all units form part of the overall sales region, i.e. lakes and other "nonselling" regions can be included also.

Table 1 provides a comparison of lower and upper bounds. Columns 1 and 2 characterize the instance class, i.e. problem size under consideration in each row in terms of $|J|$ and $|I|$, respectively. Columns 3 and 4 report the results which have been obtained using the LP-solver of CPLEX (cp. Bixby and Boyd 1996). More specifically, Column 3 provides the average upper bound UB, which has been obtained by solving the LP-relaxation of the linearized version. Column 4 shows the average CPU-time in seconds required to compute UB. Recall that averages over ten instances for each row, i.e. instance class (J, I), are provided. Columns 5 to 7 with the header CONIMP present the results of the Procedures Construct and Improve. More specifically, LB cites the average best feasible solution, i.e., lower bound computed. CPU denotes the average CPU-time in seconds required by the algorithms to compute LB. $\ll 1$ denotes that the average is only an ϵ above zero seconds. Finally, GAP $= (\text{UB} - \text{LB})/\text{UP} \cdot 100$ measures the average percentage deviation between upper and lower bound, i.e., the solution gap. Note that GAP covers both the tightness of the LP-relaxation and the deviations of the lower bounds obtained from the (unknown) optimal objective function values. On average, the solution gap roughly equals 3%. Hence, the feasible solution computed indeed must be very close to the optimal one.

Now the question shall be answered as to which of the Procedures—Construct or Improve—contributes to what extent to the fact that the lower bounds are very close to the optimum. Table 2 provides an answer. The header CON groups the data provided with respect to the Procedure Construct while the header IMP does so for the Procedure Improve. In the former case LB denotes the lower bound obtained, while in the latter case it shows the additional improvement. In both cases CPU denotes the required CPU-time in seconds. API provides the average percentage improvement.

Some important facts should be emphasized regarding the results reported in Tables 1 and 2: (i) Roughly speaking, the solution gap decreases from 4% to 2% while the size of the instance increases because of two reasons. First, relaxing the connectivity requirements makes the LP-bounds for small problem instances

Table 2 Comparison of the Procedures Construct and Improve

		CON		IMP		
$\|J\|$	$\|I\|$	LB	CPU	LB	CPU	API
50	10	11,454.97	$\ll 1$	11,508.89	$\ll 1$	0.47
	25	12,190.75	0.90	12,217.26	$\ll 1$	0.22
	50	12,724.97	2.60	12,736.17	$\ll 1$	0.09
100	10	25,173.43	$\ll 1$	25,610.99	0.50	1.73
	25	27,528.56	4.00	27,619.47	0.10	0.33
	50	28,416.35	13.40	28,464.76	0.10	0.17
250	50	69,454.64	132.50	69,774.65	10.00	0.46
500	50	129,746.85	544.70	130,962.26	82.00	0.94

weak compared to large ones. Second, the quality of the piecewise linear approximation increases with increasing problem size and, hence, makes the LP-bounds tighter. (ii) The larger the cardinality of the set I the more time has to be spent in evaluating the size and the location of the sales force. Clearly, this takes more CPU-time the larger I is in relation to J. From another point of view, if there is no degree of freedom with respect to the size of the sales force and the location of the salesmen, i.e., $\underline{S} = \bar{S}$, then the alignment and the allocation subproblems are also solved very effectively and very efficiently by our algorithms. (iii) In general, the quality of the solutions computed by the Procedure Construct is already so good that only minor improvements can be obtained subsequently. In other words, exploiting the degree of freedom on the level of the sizing and the locating decisions appropriately already gives an overall sales force deployment that can hardly be improved by realigning some of the SCUs.

The scope of the experiment conducted so far was to show how well our algorithms work. This can only be done seriously with respect to the optimal objective function or at least an upper bound. Therefore, the experiment was limited to include only instances of the size for which the LP-relaxation of the MIP-model can be solved in reasonable time. Clearly, there is no obstacle to prevent the use of the algorithms on larger instances which might become relevant e.g. in a global marketing context. The CPU-times required by our procedure show that for really huge instances comprised of thousands of SCUs it is possible to compute near-optimal solutions within a few hours of computation.

6. Insights for Marketing Management

In what follows we will discuss managerial implications of our findings. More precisely, we will state some insights and subsequently assess their validity on the basis of experiments.

Insight 1. *The results are robust with respect to Incorrect Parameter estimates.* To evaluate Insight 1, we took one of the instances with $|J| = 250$ SCUs and $|I| = 50$ potential locations. Now assume that $b_j = 0.3$ and the $c_{i,j}$, which are generated along the lines described in §4.2, $\forall j \in J$ and $i \in I$ are the unknown, but "true" values of the parameters of the sales response function. The parameters $\hat{b}$ and $\hat{c}_{i,j}$, which are used in the experiment, are then generated via data perturbation as follows: Calculate $\hat{b} = b + \Delta b$ and choose $\hat{c}_{i,j} \in [c_{i,j}(1 - \Delta c), c_{i,j}(1 + c_{i,j})]$ at random where Δc and Δb are perturbation control parameters. Table 3 presents the results of this study. Across rows and columns we provide the percentage decrease DEC $= (\text{OPT} - \text{ACT})/\text{OPT} \cdot 100$ of profit where OPT denotes the "optimal" objective value which has been calculated based upon the "true" parameter values, while ACT is the one which has been computed with respect to the perturbed parameters. The results show that even in the case when the parameters are estimated very "badly" (i.e. all of them are under- or overestimated drastically) the percentage decrease of profit hardly exceeds 3%.

Table 3 **Robustness of the Model**

$\hat{b}$	0.20	0.25	0.30	0.35	0.40
Δb	−0.10	−0.05	0.00	0.05	0.10
$\Delta c = 0.00$	1.89	1.67	0.00	0.17	0.25
$\Delta c = 0.05$	2.25	1.92	0.36	0.87	1.25
$\Delta c = 0.10$	2.87	3.06	1.58	1.97	2.07

Insight 2. *Profit is not that sensitive with respect to sales force size.* To evaluate Insight 2, once more we took the instance with $|J| = 250$ SCUs and $|I| = 50$ potential locations. Then, the size of the sales force was set to the levels 29, 30, 31, 32, 33 by fixing $\underline{S} = s = \bar{S}$, accordingly. Table 4 provides the results of this experiment. OFV(s) denotes the objective function value (normalized to the interval [0, 1]), which has been computed by our methods with respect to the size s. The results, which are typical for various other experiments not documented here, support Insight 2. This means that the objective function is fairly flat near the optimum number of salesmen. Hence, the "flat maximum principle" (cp. Chintagunta 1993) is valid also in this context.

Table 4 **Profit as Function of Sales Force Size**

s	OFV(*s*)
29	0.99577
30	0.99919
31	1.00000
32	0.99996
33	0.99941

Table 5 **Profit as Function of Location of Salesmen**

s	OFV(*s*)—Selected Iterations					
29	0.934	0.960	0.947	0.973	0.986	1.000
30	0.938	0.927	0.957	0.934	0.977	1.000
31	0.928	0.921	0.952	0.950	0.949	1.000
32	0.933	0.924	0.965	0.967	0.962	1.000
33	0.917	0.945	0.933	0.948	0.978	1.000

Insight 3. *Profit is sensitive with respect to the location of the salesmen*. Once more we relate to the instance already used twice. Table 5 provides part of the protocol of a run. More precisely, the outcome of some typical iterations of the Procedure Locate where potential locations are evaluated systematically is given in terms of normalized objective function values OFV(s). Similarly to Table 4, the size of the sales force is fixed in each row. Clearly, the process converges to the best found objective function value (hence, OFV(s) = 1 in Column seven), but the values go up and down depending on the specific old and new locations under investigation. Hence, the "flat maximum principle" is not valid with respect to the location of the salesmen.

The insights evaluated in Tables 3, 4, and 5 can be summarized as follows: (i) For reasonable problem parameters, the size of the sales force does not affect a firm's profit that much. (ii) The location of the salesmen will, in general, affect a firm's profit drastically. Consequently, existing alternatives must be evaluated. (iii) Fortunately, the model is very robust with respect to the estimation of the parameters of the sales response function. Even in the case when there is a systematic estimation bias (over or underestimation of *all* the parameters) the decision is not that bad in terms of a firm's profit. Usually, there is no systematic bias, hence, the sales force deployment evaluated by the algorithms will be superb.

7. Application

The methods presented in this paper have been used to redesign the sales force deployment of the company "FAXE Getränke–Vertriebs GmbH," a major distributor of beverages, especially beer. FAXE is part of the Danish distributor of beverages "Bryggerigroupen Danmark AS."

In recent years, FAXE has intensified the sales of beverages via large discount stores, while at the same time decreasing its presence in small stores. Because of the changed distribution policy, the management decided to redesign the sales force deployment. More specifically, the aim was to reduce the sales force size while increasing the selling time of each salesman, which apparently can only be achieved by relocating the salesmen. Previous efforts to increase profits were mainly concerned with realigning SCUs.

The purpose of the application was to redesign the sales force deployment of FAXE in Schleswig–Holstein (northern part of Germany). Some characteristics of the application are as follows: FAXE has a turnover of 15 million DM per year from beer. Schleswig-Holstein has 2.7 million inhabitants. Furthermore, there are 1,219 SCUs (statistical districts). In total, Schleswig-Holstein has 4,000 stores that sell beverages. Only those 650 of the stores located in SCUs that have at least 4,000 inhabitants are served by FAXE. Currently, FAXE employs five salesmen in Schleswig-Holstein, each of which costs the company DM 120,000 per year on average (salary and car). Each salesman works from 7 a.m. to 5 p.m. (10 hours selling time per day) where on average 50% is effective calling time.

The sales territory under concern, i.e., the 1,219 SCUs, are represented by polygon lines. Two SCUs are adjacent if they share at least one line of the polygon. The adjacency structure of the application is pretty much the same as the one reflected by the data generator; it reveals 5.73 neighbors on average. Travel times are estimated on the basis of the total road network of Schleswig-Holstein, despite inner-city streets. Geographic barriers between adjacent SCUs result in high travel times. Consider, for instance, the canal which links the Baltic and the North Sea. Surmounting this barrier, e.g. by ferry, takes quite a long time although the Euclidean distance between two SCUs is very small. Obviously, high travel times decrease the probability that two such SCUs are assigned to the same sales territory center. Finally, it should be noted that the assumptions mentioned in §3 with regard to Equations (2) and (3) are satisfied to a very large extent in this case.

A summary of the parameters in terms of what has been presented in the previous sections can be given as follows: $|I| = 125$, $\underline{S} = 3$, $\bar{S} = 8$, $|J| = 1{,}219$, $T_i = 1{,}300$, $b = 0.285$, $c_{i,j} = 0.205 \cdot H_j \cdot (4.6 - d_{i,j})^b$ where H_j denotes the number of inhabitants of SCU j, and $d_{i,j}$ estimates the time to drive from SCU i to SCU j.[2]

The methods described previously have been applied to the FAXE case. The approximation methods run 170 sec on a 133 MHz Pentium machine. The results can be summarized as follows: Staying with the current number of five salesmen, decrease of profit can be stopped to some extent only by relocating three of them. The solution computed by means of our approximation methods, however, demands, that the sales force size be decreased from 5 to 4, where each of the salesmen has to be relocated. More specifically, we have $Z(3) = 1{,}222{,}694$, $Z(4) = 1{,}245{,}021$, and $Z(5) = 1{,}198{,}821$, respectively; that is, the sales contribution as a function of the sales force size 3, 4, and 5 is concave. The result is displayed graphically in Figure 1 where the four sales territories are put in shading. The new sales territory centers have to be located in SCUs $\mathcal{I} = \{50, 56, 74, 521\}$, indicated by "•". Note that SCU $j = 50$ (56, 74, 521) has 11,793 (8,254, 28,524, 8,211) inhabitants. We also let our upper bounding procedure run where the number of sales territory centers was fixed in advance ($\underline{S} = \bar{S} = 3$, 4, and 5), thus keeping the MIP-model of Appendix A manageable. We obtained upper bounds, which are in the percent range presented in §5, and they are UB(3) < UB(4) > UB(5); i.e., the upper bounding function is concave also. Hence, our proposal of four sales territory centers is supported with regard to upper bounds, as well, although the optimal solution is unavailable. Currently, the management of FAXE is going to implement the new sales territory deployment. It is expected that the profits will increase in the years to come.

[2] Some of the data are biased because FAXE views our work with them as proprietary. They do not want their competitors to know all the details.

8. Summary and Conclusions

In this paper we show how four interrelated sales force deployment subproblems can be modelled and solved simultaneously. These subproblems are sizing the sales force, salesman location, sales territory alignment, and sales resource allocation. More specifically, an integrated nonlinear mixed-integer programming model is formulated. For the solution of the model we present a newly developed effective and efficient approximation method.

The methods are evaluated on two sets of instances. The first one stems from a systematic generation of a representative set of problem instances covering all problem parameters at hand. The second one is an application in the distribution of beverages. Benchmarking the results with the help of upper bounds shows that the method allows very fast solution of large-scale instances close to optimality.

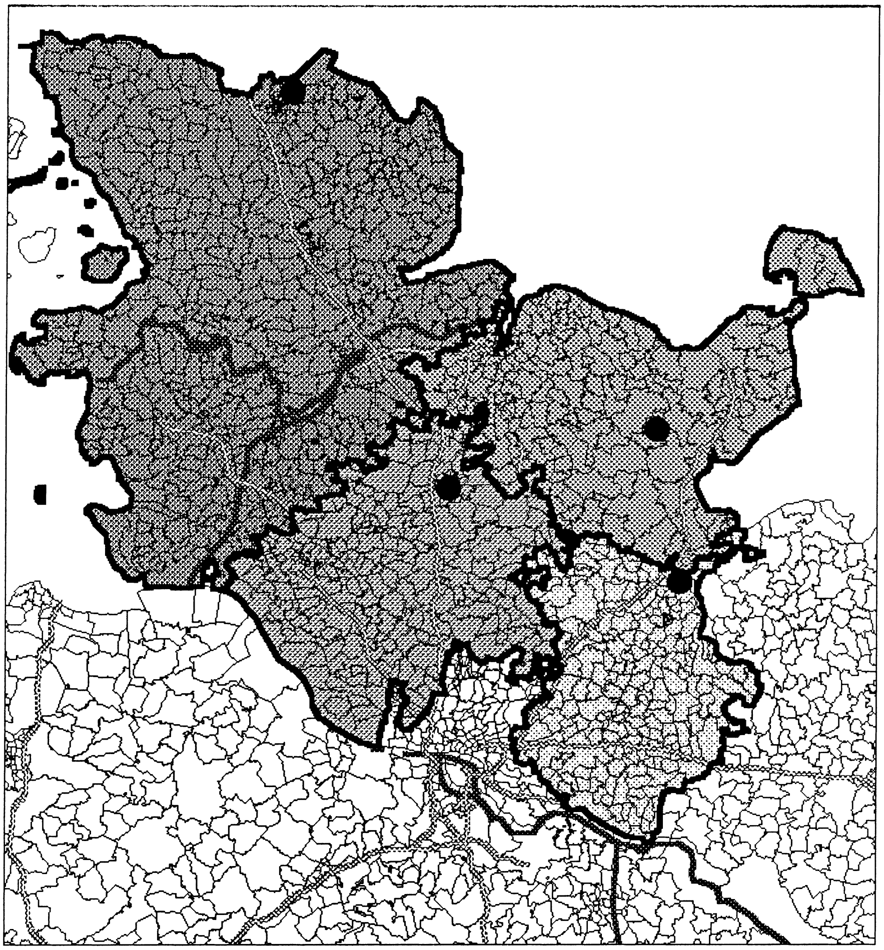
The methods which provide lower bounds for the optimal objective function value are benchmarked against upper bounds. On average the solution gap, i.e. difference between upper and lower bound, is roughly 3%. Furthermore, it is shown, how the methods can be used to analyze various problem settings which are of high practical relevance.[3]

[3] This work has been inspired by the research of Sönke Albers and Bernd Skiera. Moreover, the support of Harald Behre, Distribution Manager of FAXE, is gratefully acknowledged. Finally, the critical comments of three anonymous reviewers and the associate editor helped to improve the presentation.

Appendix A: Mixed-Integer Linear Programming Model

The purpose of the following is to show how the nonlinear objective function (4) can be approximated by a piecewise linear curve in order to get a model formulation which is accessible to general

Figure 1 Redesigned Sales Territory for Schleswig—Holstein



purpose MIP-solvers. Practically, this is done as follows (cp. Haase 1997 also): Approximate each nonlinear term by a piecewise linear curve, as pictured in Figure 2, which provides the functional relationship between $t_{i,j}$ and $S_{i,j}$ (solid line). In Figure 2 the function $S_{i,j}$ is approximated by three segments. The constraints $0 \le t_{i,j} \le T_i$ imply that we need not extend the approximation beyond these bounds on the variables. The dashed approximation curves are determined by linear approximations between breakpoints. At time instant $\tilde{t}_{i,j,\tau}$ the sales response function $S_{i,j}$ equals $G_{i,j,\tau}$, which is defined by Equation (15):

$$G_{i,j,\tau} = \alpha_{i,j,\tau} + \beta_{i,j,\tau} t_{i,j}. \tag{15}$$

The definition of $G_{i,j,\tau}$ is based on the observation that the tangent and the sales response function have identical gradients at time instant $\tilde{t}_{i,j,\tau}$. The constant term $\alpha_{i,j,\tau}$ and the gradient $\beta_{i,j,\tau}$ are defined by Equations (16) and (17):

$$\alpha_{i,j,\tau} = c_{i,j}(\tilde{t}_{i,j,\tau})^{b_j} - \beta_{i,j,\tau}\tilde{t}_{i,j,\tau}, \tag{16}$$

$$\beta_{i,j,\tau} = c_{i,j} b_j (\tilde{t}_{i,j,\tau})^{b_j - 1}. \tag{17}$$

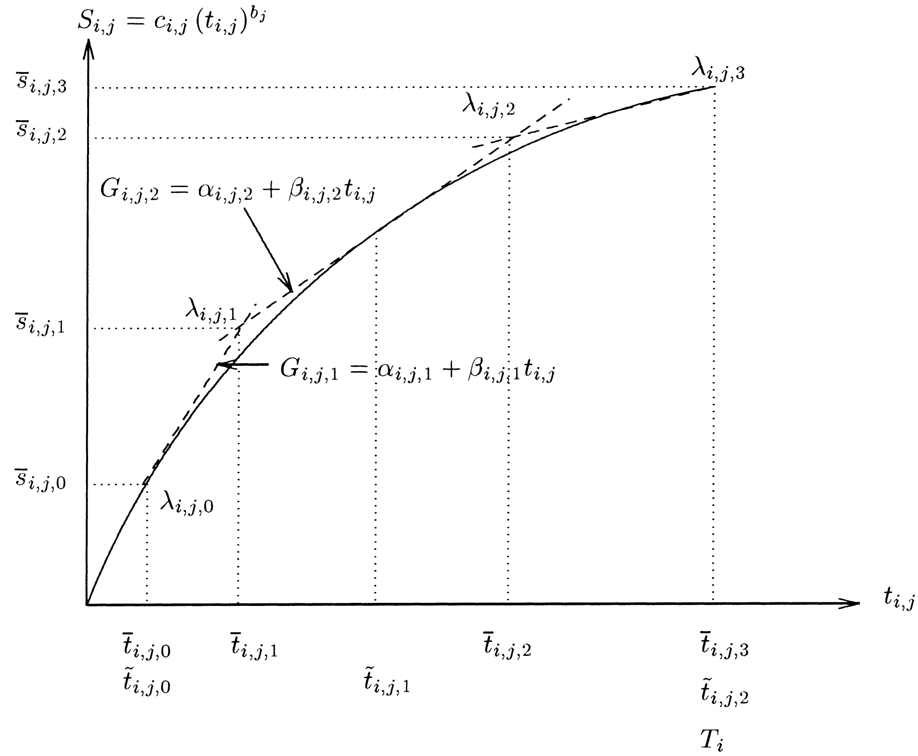
Note that only $\tilde{t}_{i,j,\tau}$ has to be predefined in order to calculate $\alpha_{i,j,\tau}$ and $\beta_{i,j,\tau}$. To fully describe the approximation, the intersection points $(\bar{t}_{i,j,\tau}, \bar{s}_{i,j,\tau})$ have to be calculated. At time instant $\bar{t}_{i,j,\tau}$ we observe $G_{i,j,\tau-1} = G_{i,j,\tau}$, hence Equation (18) is valid:

$$\bar{t}_{i,j,\tau} = \frac{\alpha_{i,j,\tau+1} - \alpha_{i,j,\tau}}{\beta_{i,j,\tau} - \beta_{i,j,\tau+1}}, \tag{18}$$

and $\bar{s}_{i,j,\tau}$ is calculated similarly. Now, the interval $[\bar{t}_{i,j,\tau}, \bar{t}_{i,j,\tau+1}]$ describes the relevant part of the time axis for which the curve $G_{i,j,\tau}$ must be defined.

Let $\lambda_{i,j,\tau} \in [0, 1]$ denote a weight for the point $(\bar{t}_{i,j,\tau}, \bar{s}_{i,j,\tau})$. Then the curve $G_{i,j,\tau}$ corresponds to a linear combination of the points $(\bar{t}_{i,j,\tau}, \bar{s}_{i,j,\tau})$ and $(\bar{t}_{i,j,\tau+1}, \bar{s}_{i,j,\tau+1})$ where the weights $\lambda_{i,j,\tau}$ and $\lambda_{i,j,\tau+1}$ satisfy $\lambda_{i,j,\tau} + \lambda_{i,j,\tau+1} = 1$. At time instant $\lambda_{i,j,\tau}\bar{t}_{i,j,\tau} + \lambda_{i,j,\tau+1}\bar{t}_{i,j,\tau+1}$ an approximation of the expected sales is given by $\lambda_{i,j,\tau}\bar{s}_{i,j,\tau} + \lambda_{i,j,\tau+1}\bar{s}_{i,j,\tau+1}$.

Figure 2 **Linearization of the Objective Function**



The weights $\lambda_{i,j,\tau}$ must satisfy the following *neighborhood adjacency* restriction (cp. Bradley et al. 1977): At most two $\lambda_{i,j,\tau}$ weights are positive. If two weights are positive, then they are adjacent, i.e. of the form $\lambda_{i,j,\tau}$ and $\lambda_{i,j,\tau+1}$. It is essential to note that, for *concave* objective functions, the adjacency condition will always be enforced by the maximization and hence can be ignored. As a consequence the approximation of the objective function can be done without any additional integer variables.

We summarize the newly introduced parameters

$\bar{t}_{i,j,\tau}$: τ-th time instant for subdividing the maximal workload T_i of salesman i for SCU j, i.e. the selling time;

$\bar{s}_{i,j,\tau}$: approximation of the expected sales when SCU j is aligned to sales territory i and the selling time is $\bar{t}_{i,j,\tau}$;

$P_{i,j}$: ordered set $(0, \ldots, |P_{i,j}| - 1)$ used to index the τ-th time instant $\bar{t}_{i,j,\tau}$ and the τ-th sales contribution $\bar{s}_{i,j,\tau}$.

We use the already defined variables $x_{i,j}$, define the variables

$$\lambda_{i,j,\tau}\text{: weight of the point } (\bar{t}_{i,j,\tau}, \bar{s}_{i,j,\tau}),$$

and get the following mixed-integer linear programming model (MP):

$$\text{maximize } Z_{\mathrm{MP}}(x, \lambda) = \sum_{i\in I}\sum_{j\in J}\sum_{\tau\in P_{i,j}} \bar{s}_{i,j,\tau}\lambda_{i,j,\tau} - \sum_{i\in I} f_i x_{i,i} \tag{19}$$

subject to (7), (8), (9) and

$$\sum_{\tau\in P_{i,j}} \lambda_{i,j,\tau} = x_{i,j} \quad (i \in I,\ j \in J), \tag{20}$$

$$\sum_{j\in J}\sum_{\tau\in P_{i,j}} \bar{t}_{i,j,\tau}\lambda_{i,j,\tau} = T_i x_{i,i} \quad (i \in I), \tag{21}$$

$$\lambda_{i,j,\tau} \geq 0 \quad (i \in I,\ j \in J,\ \tau \in P_{i,j}). \tag{22}$$

The linear term (19) maximizes sales while taking the fixed cost of the salesman locations into account and hence maximizes profit, similarly to its nonlinear counterpart (4). Equation (20) aligns SCU j to sales territory i if the salesman i visits SCU j. Similarly to Equation (6), Equation (21) guarantees that the maximum workload per period and salesman is regarded. Having "$\leq$" instead of "$=$" in Equation (21) would be fine, too. Equations (9) and (22) define the decision variables appropriately. Apparently, MP is a mixed-integer linear programming model which can be solved by general purpose MIP-solvers.

Finally, some remarks shall be given: (i) Apparently, for time instants $t_{i,j} = \lambda_{i,j,\tau}\bar{t}_{i,j,\tau} + \lambda_{i,j,\tau+1}\bar{t}_{i,j,\tau+1}$ expected sales are overestimated. For time instant $\tilde{t}_{i,j,\tau}$ both curves coincide. As a consequence, the inequality $Z^*_{\mathrm{NLP}} \leq Z^*_{\mathrm{MP}}$ holds where Z^*_{NLP} defines the optimum objective function value of the model NLP and Z^*_{MP} is the optimum objective function value of the model MP, respectively. (ii) The relationship between an optimal resource allocation $\bar{t}^*_{i,j,\tau}$ of the model MP and a feasible resource allocation $t_{i,j}$ of the model NLP is defined by

Table 6 Procedure Construct

```
Initialize s̲ = S̲, s̄ = S̄, LB = −∞, lose[h, i] = 0, h ∈ I, i ∈ I
WHILE s̲ ≤ s̄ DO
  s = ⌊(s̲ + s̄)/2⌋
  set |I_1| = s, I_1 = {i_1, . . . , i_s}
  call Procedure Locate (s)
  call Procedure Align (L(I_1))
  evaluate resources allocation by Equation (13) or (14)
  IF Z(s + 1, x, t) > Z(s, x, t) THEN
    s̲ = s + 2
  ELSE
    s̄ = s − 1
  ENDIF
  IF Z(s + 1, x, t) > LB THEN
    LB = Z(s + 1, x, t)
    |𝒥| = s + 1
  ENDIF
  IF Z(s, x, t) > LB THEN
    LB = Z(s, x, t)
    |𝒥| = s
  ENDIF
ENDWHILE
```

Table 7 Procedure Locate (s)

```
Initialize |I_1| = s, I_0 = I\I_1, L(I_1), improve = TRUE
WHILE improve DO
  improve = FALSE
  FOR k = 1 TO |I_1| DO
    h = min{i ∈ I_0|lose[i_k, i] ≤ lose[i_k, g] ∀ g ∈ I_0}
    I_1 = I_1\i_k ∪ h
    update I_0 and L(I_1)
    IF Z(L(I_1)) > LB THEN
      improve = TRUE
      LB = Z(L(I_1))
    ELSE
      I_1 = I_1 ∪ i_k\h
      update I_0 and L(I_1)
      lose[i_k, h] = lose[i_k, h] + 1
    ENDIF
  ENDFOR
ENDWHILE
```

Table 8 Procedure Align $(L(I_1))$

```
Initialize J_0, J_i, C_i and 𝒜_i
WHILE J_0 ≠ ∅ DO
  compute (h, i) such that c_{h,i}/C_i ≥ c_{k,j}/C_j ∀ i ∈ I_1, ∀ j ∈ I_1, ∀ h
      ∈ 𝒜_i, ∀ k ∈ 𝒜_j
  J_0 = J_0\h
  J_i = J_i ∪ h
  C_i = C_i + c_{h,i}
  update 𝒜_i
ENDWHILE
```

the equation $t_{i,j} = \sum_{\tau \in P_{i,j}} \lambda_{i,j,\tau} \bar{t}^*_{i,j,\tau}$. (iii) On the other hand, if we express an optimum solution of MP in terms of NLP the former produces a lower bound LB of the optimum objective function value for the latter. More formally, this is expressed by the inequality $\mathrm{LB} \le Z^*_{\mathrm{NLP}}$. (iv) The quality of the approximation is affected by the number and the location of the tangents. Clearly, it is not necessary that the intervals $[\bar{t}_{i,j,\tau}, \bar{t}_{i,j,\tau+1}]$ are equidistant. They can be chosen with respect to the maximum tolerated approximation error.

Appendix B: Formal Description of the Algorithms

A formal description of the procedures provided in §§4 and 5 can be found in Tables 6 thru 10.

Appendix C: Special Cases and Extensions

The problem setting investigated in this paper is already very general. In the following we will point to some special cases covered by the model formulation MP in a formal way. In addition, an extension will be given.

(i) A lower bound $\underline{S}$ and an upper bound $\bar{S}$ on the number of salesmen (i.e. the sales force size) will be respected by adding the constraints (23) and (24):

$$\sum_{i \in I} x_{i,i} \ge \underline{S}, \tag{23}$$

$$\sum_{i \in I} x_{i,i} \le \bar{S}. \tag{24}$$

Clearly, our model reduces to the special case considered by Skiera and Albers (1994, 1996) for $\underline{S} = \bar{S}$.

(ii) In some applications it might be necessary to demand a minimum and/or a maximum selling time of salesman i with respect to SCU j. Predefining $\bar{t}_{i,j,0}$ and $\bar{t}_{i,j,|P_{i,j}|-1}$ appropriately allows taking care of such requirements easily. If $\bar{t}_{i,j,0}$ is relatively large $\forall i \in I$ then it might be appropriate to assign SCU j to none of the sales territories. In this situation we replace the equality "=" in Equation (7) by the inequality "≤".

(iii) If the number of SCUs which might be aligned to salesman i should be bounded from above then we add the constraint

$$\sum_{j \in J} x_{i,j} \le U_i \tag{25}$$

to the already existing constraint set where U_i defines the maximum number of SCUs.

(iv) As already mentioned balancing sales territories with respect to some attributes has been the scope of interest of several researchers. Let u_j denote the sales potential, i.e. weight of SCU j. Further assume that $|I|$ locations have already been established, i.e. $x_{i,i} = 1 \; \forall i \in I$. If we have to come up with a balanced sales force deployment then we have to replace the constraints (20) and (21) by constraint (26):

Table 9 Procedure Improve (J)

```
Initialize improve = TRUE, LB = Z(J)
WHILE improve DO
  improve = FALSE
  FOR j = 1 TO |𝒥| DO
    FOR i = 1 TO |I_1| DO
      IF add(J_i, j) ∧ drop(J_i, j) ∧ LB < Z(J|J_{i(j)}\j, J_i ∪ j) THEN
        LB = Z(J|J_{i(j)}\j, J_i ∪ j)
        J_{i(j)} = J_{i(j)}\j
        J_i = J_i ∪ j
        improve = TRUE
      ENDIF
    ENDFOR
  ENDFOR
ENDWHILE
```

Table 10 Procedure Generate (I, J)

```
Initialize ℳ = {(Q, Q)} and 𝒜 = 𝒩_(Q,Q)
WHILE |ℳ| ≤ |J| DO
  Choose (α, β) ∈ 𝒜 at random
  ℳ = 𝒩 ∪ (α, β)
  𝒜 = 𝒜 ∪ 𝒩_(α,β) ∩ ℳ
ENDWHILE
𝒩_(α,β) = 𝒩_(α,β) ∩ 𝒜 ∀(α, β) ∈ ℳ
choose I ⊆ J sales territories at random
```

$$\sum_{j\in J} u_j x_{i,j} - \delta_i^+ + \delta_i^- = \frac{1}{|I|}\sum_{j\in J} u_j \quad (i \in I). \tag{26}$$

Furthermore, the objective (19) has to be replaced by

$$\text{minimize} \sum_{i\in I} \delta_i^- + \delta_i^+ \tag{27}$$

and the nonnegativity constraints (28) have to be added to the constraint set

$$\delta_i^-, \delta_i^+ \geq 0 \quad (i \in I). \tag{28}$$

The variables δ_i^+ and δ_i^- achieve a balance of sales territory i with respect to the average weight $(1/|I|)\sum_{j\in J} u_j$.

Notice that modifying the set of equations of MP implies that the MIP-solution methodology remains applicable in all four cases. Furthermore, case (i) is already covered by our approximation methods while cases (ii) and (iii) only need minor modifications. To solve case (iv) heuristically also would require substantial changes of the methods presented in this paper.

References

Albers, S. 1989. *Entscheidungshilfen für den Persönlichen Verkauf*. (in German). Duncker & Humblot, Berlin, Germany.

——, M. Krafft. 1992. Steuerungssysteme für den Verkaufsaußendienst. (in German). Technical Report 306, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, Kiel, Germany.

Beckmann, M. J., T. F. Golob. 1972. A critique of entropy and gravity in travel forecasting. F. N. Gordon, ed. *Traffic Flow and Transportation*. New York, 109–117.

Beswick, C. A. 1977. Allocating selling effort via dynamic programming. *Management Sci.* **23** 667–678.

Bixby, N., E. Boyd. 1996. *Using the CPLEX Callable Library*. CPLEX Optimization Inc., Houston, TX.

Bradley, S. P., A. C. Hax, T. L. Magnanti. 1977. *Applied Mathematical Programming*. Addison-Wesley, Reading, MA.

Chintagunta, P. K. 1993. Investigating the sensitivity of equilibrium profits to adverstising dynamics and competitive effects. *Management Sci.* **39** 1146–1162.

Churchill, G. A., N. M. Ford, O. C. Walker. 1993. *Sales Force Management*. 4th ed. Irwin, Homewood, IL.

Einbu, J. M. 1981. Extension of the Luss-Gupta resource allocation algorithm by means of first order approximation techniques. *Oper. Res.* **29** 621–626.

Glaze, T. A., C. B. Weinberg. 1979. A sales territory alignment program and account planning system (TAPS). R. P. Bagozzi, ed. *Sales Management: New Developments from Behavioral and Decision Model Research*. Marketing Science Institute, Cambridge, MA. 325–343.

Haase, K. 1997. Deckungsbeitragsorientierte Verkaufsgebietseinteilung und Standortplanung für Außendienstmitarbeiter. (in German.) *Z. betriebswirtschaftliche Forschung* **49** 877–891.

Heschel, M. S. 1977. Effective sales territory development. *J. Marketing* **41**(2) 39–43.

Lodish, L. M. 1980. A user-oriented model for sales force size, product, and market allocation decisions. *J. Marketing* **44** 70–78.

Mantrala, M. K., P. Sinha, A. A. Zoltner. 1992. Impact of resource allocation rules on marketing investment-level decisions and profitability. *J. Marketing Res.* **29** 162–175.

Segal, M., D. B. Weinberger. 1977. Turfing. *Oper. Res.* **25** 367–386.

Shanker, R. J., R. E. Turner, A. A. Zoltners. 1975. Sales territory design: An integrated approach. *Management Sci.* **22** 309–320.

Skiera, B., S. Albers. 1994. COSTA: Ein Entscheidungs-Unterstützungs-System zur deckungsbeitragsmaximalen Einteilung von Verkaufsgebieten. (in German.) *Z. Betriebswirtschaft* **64** 1261–1283.

——, ——. 1996. COSTA: Contribution optimizing sales territory alignment. Technical Report 408, Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel, Germany.

——, ——. 1998. COSTA: Contribution optimizing sales territory alignment. *Marketing Sci.* **17** 196–213.

Zoltners, A. A., P. Sinha. 1980. Integer programming models for sales resource allocation. *Management Sci.* **26** 242–260.

——, ——. 1983. Sales territory alignment: A review and model. *Management Sci.* **29** 1237–1256.

——, ——, P. S. C. Chong. 1979. An optimal algorithm for sales representative time management. *Management Sci.* **25** 1197–1207.

Accepted by Dipak Jain; received November 18, 1996. This paper has been with the authors 1 year and 2 months for 2 revisions.