

Vetschera, Rudolf

**Working Paper**

## Composite alternatives in group decision support

Diskussionsbeiträge - Serie I, No. 262

**Provided in Cooperation with:**

Department of Economics, University of Konstanz

*Suggested Citation:* Vetschera, Rudolf (1992) : Composite alternatives in group decision support, Diskussionsbeiträge - Serie I, No. 262, Universität Konstanz, Fakultät für Wirtschaftswissenschaften und Statistik, Konstanz

This Version is available at:

<https://hdl.handle.net/10419/68934>

**Standard-Nutzungsbedingungen:**

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

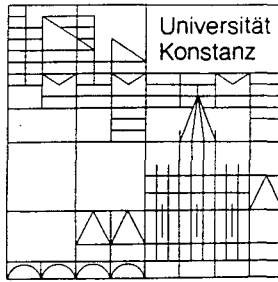
Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.

**Terms of use:**

*Documents in EconStor may be saved and copied for your personal and scholarly purposes.*

*You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.*

*If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.*



## **Fakultät für Wirtschaftswissenschaften und Statistik**

---

Rudolf Vetschera

### **Composite Alternatives in Group Decision Support**

19. JAN. 1993 Weltwirtschaft  
Kiel

10 284(262) *h*

**Diskussionsbeiträge**

---

Postfach 5560  
D-7750 Konstanz

Serie I — Nr. 262  
Dezember 1992

# **Composite Alternatives in Group Decision Support**

**Rudolf Vetschera**

**Serie I - Nr. 262**

**November 1992**

# **COMPOSITE ALTERNATIVES IN GROUP DECISION SUPPORT\***

**Rudolf Vetschera**

1992-11-26

## **Abstract**

Decision theoretic models of group decision processes usually assume a given set of alternatives, about which the decision has to take place. In realistic group decision situations, however, alternatives are often not specified a priori, but are created during the group process from different components introduced by the group members. This paper develops methods for systematically creating such "composite alternatives", taking also into account the necessity to keep both the computational effort and the cognitive load to group members within reasonable limits.

---

\* Paper presented at the International Conference on Support Systems for Decision and Negotiation Processes DNS-92 , Warsaw 1992

## 1. Introduction

Many group decision support systems presented in the literature are based on the assumption that the group decision problem consists of choosing among a predefined, fixed number of alternatives (e.g. SCDAS: Lewandowski et al., 1987; Co-oP: Bui, 1987, GDSS-X: Vetschera, 1991b). An example which is often used to illustrate this kind of problem is the selection of one candidate out of several applicants for a given position. In this setting, group decision support mainly consists in unifying and aggregating the individual evaluations of alternatives by the group members.

In many other problems, however, the definition of alternatives is not as clear as in this example. Alternatives are not given a priori, but comprise different actions performed by the group members. For example, in a corporate context, an investment board might contain as its members the heads of separate divisions of the company. Each division has one or more proposals for investments which can be carried out in that division. The decision problem, which the group as a whole has to solve, is to formulate an investment program for the entire company, consisting of several individual projects. In such a situation, the definition of alternatives itself becomes part of the problem.

This subproblem, in turn, depends on the context of the entire group decision problem. In this paper, we view a group decision problem basically as a decision problem with several criteria. Group members might differ in the criteria they use for evaluating alternatives and the importance they assign to those criteria. They also might view the performance of alternatives in those criteria differently. Other sources of problem complexity and differences between group members, as for example risk and different risk attitudes of members, will not be considered explicitly in this paper. They can, however, easily be introduced into the general framework developed here.

In a multi-criteria group decision situation, the individual control which members exercise over different aspects of the problem might have two kinds of effects: an individual action, which a group member can take within the joint decision might either concern only criteria in which only that member is interested, or criteria which are also relevant for some other members. In the first case, the member can simply apply whatever changes he/she finds appropriate to the choice made by the group, without interfering with the other members' interests. The second case causes specific problems for group decision support and will be the topic of this paper.

In this case, we can formally structure the decision problem as a choice among alternatives, which are themselves combinations of different individual actions by group members. We will call such alternatives "composite alternatives" in order to distinguish them from the a

priori defined alternatives which are usually considered. Section two of this paper develops this concept in detail and reviews related approaches proposed in the literature.

The concept of composite alternatives raises two closely connected problems: the first problem is that of aggregating individual components to actual decision alternatives, which the group as a whole faces. A basic approach for this generation process will be developed in section three. Since this generation is essentially a combinatorial problem, a second problem arises as the potential number of alternatives becomes very large and techniques for the reduction of the set of alternatives have to be developed. Possible strategies for such a reduction in problem complexity will be developed in section four. Section five concludes the paper by providing an outlook on future research directions.

## **2. Composite Alternatives**

Composite alternatives can arise in both cooperative and non-cooperative group decision problems. Examples of composite alternatives in cooperative settings are joint programs of actions, which consist of different measures undertaken by different group members. In non-cooperative settings, composite alternatives can be seen as "packages" of concessions, which the different parties will jointly undertake.

The entire decision process as envisioned in this paper is represented in figure 1. Here, each group member has control over specific aspects of the group decision problem. We call the actions which each group member might undertake "Partial Actions" (PA). Alternatives at the group level are constructed as combinations of several of the proposed partial actions. We will therefore call them "Composite Alternatives" (CA). To determine the consequences of each CA, an aggregation of the contributions of all partial actions involved is necessary. The resulting CAs at the group level are then subject to the actual group decision process, which is represented by the shaded box in figure 1. In a dynamic perspective, this process might consist of several iterations, in which the group members individually evaluate the alternatives. Evaluations are aggregated and preliminary group results influence the individual evaluations in the next iteration. The structure of the group decision process itself will not be considered in this paper. For a survey of this area see e.g. (Vetschera, 1990).

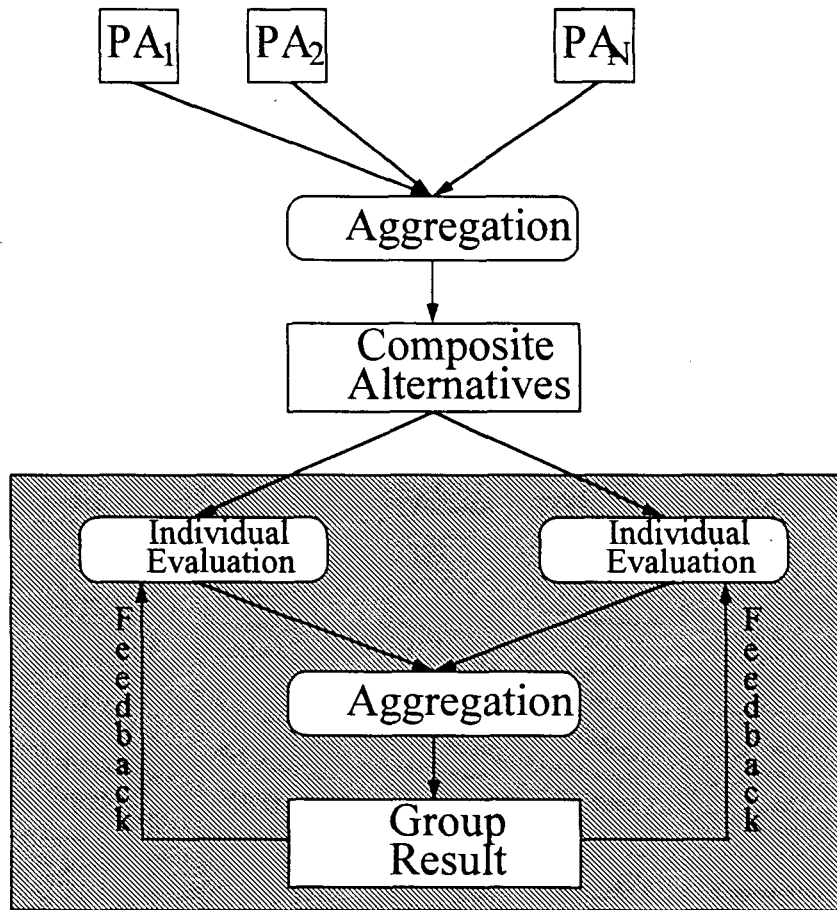


Figure 1: Decision Process

Especially in the context of negotiations between adversaries, similar concepts have already been proposed in the literature. One example is Saaty's approach to the resolution of retributive conflicts (Saaty, 1987). Saaty considers a two-party negotiation problem. Each party has the possibility to make concessions, corresponding to PAs in our framework. Mutual concessions from both parties form packages similar to CAs, which are evaluated by both parties. For party A, the following expression is used in this evaluation:

$$A's \text{ ratio} = \frac{\sum_{B's \text{ concessions}} A's \text{ benefits} \times B's \text{ costs perceived by A}}{\sum_{A's \text{ concessions}} B's \text{ benefits perceived by A} \times A's \text{ costs}}$$

A package of concessions is assumed to be acceptable for both parties, if both ratios are greater than one and no party perceives the other party's gains as considerably higher than its own.

Saaty proposes to generate all packages of concessions fulfilling these conditions and to have an outside arbitrator choose packages which seem to be fair exchanges, taking also into account, for example, different time perspectives and the importance of individual

concessions. The generation process for these packages, however, is not described in detail.

Another example, also from the area of negotiations between adversaries, is the NegoPlan system (Matwin et al., 1989; Kersten et al., 1991). In contrast to Saaty's approach, NegoPlan was developed to support only one side in a bargaining situation. The supported party's goal structure is represented as a tree of logical values, called the "Goal Representation Tree" (GRT). In NegoPlan, goals can only be fulfilled or not fulfilled, there is no partial achievement of goals. The leaves of the GRT represent certain facts, which together form the topic of the bargaining problem. For example, in a wage bargaining problem, one fact represented by a leaf might indicate whether certain benefits are granted or not. These leaves therefore closely correspond to our PAs. The top node of the tree indicates whether a party is satisfied with a certain configuration of partial actions or not. From the GRT, NegoPlan generates another tree called the "Goal Solution Tree" (GST). A GST represents one solution, and therefore one set of actions, which is acceptable for a party. It is therefore similar to the concept of a CA. From such a tree, the party learns which of the facts it must actually obtain in the bargaining process and which it can give away as "bargaining chips". Of course, several GSTs might exist for a given GRT and NegoPlan can generate different trees, which then correspond to different bargaining strategies. However, NegoPlan does not systematically generate all GSTs for a given problem. Since partial achievement of goals is not possible, the generation of all GSTs is not necessary in the NegoPlan framework. However, if one tries to take cardinal notions of preference into account, different solutions could also be evaluated (and perhaps ranked) in more detail. In this case, the generation of all possible combinations of actions would be desirable.

Other approaches to group decision support view such problems not as multi-attribute decision problems, in which a given set of alternatives is considered, but as multi-objective mathematical programming problems, in which the set of alternatives is only implicitly described via a set of constraints (Isermann, 1984; Bronisz et al., 1989). This approach has the advantage of taking into account all possible group level alternatives, even if only implicitly. If, on the other hand, all possible group level alternatives are explicitly generated from the set of PAs, it might be necessary to subsequently reduce the resulting set in size in order to reduce the complexity of the decision problem. It is possible that the best compromise will be eliminated in the process. The computational effort needed for the total process might also be smaller if not every possible group level alternative is generated. This advantage, however, might not be as large as expected, since typical multiobjective methods like the STEM method used in (Isermann, 1984) require the solution of many scalar optimization problems.



An approach in which alternatives are given only implicitly puts considerable cognitive strain on the group members. Even if they are provided with samples from the efficient frontier, they still have to imagine other solutions, perhaps just on the basis of some implicit information about possible trade-offs. It will certainly be easier for them to study and interpret a finite set of alternatives.

The use of explicit alternatives might also make it easier to reach a consensus. For many decision problems, only ordinal rankings of alternatives or merely the selection of a single "best" alternative are required (Roy, 1991). However, individual evaluations can more easily involve a cardinal evaluation of alternatives. An approach based on explicitly given alternatives makes it possible to exploit the span between ordinal and cardinal evaluations to support consensus. Examples for such methods are the N.A.I. algorithm (Bui, 1985) or the feedback-oriented approach (Vetschera, 1991a).

### 3. Generation Techniques

#### 3.1. Data Requirements

As already indicated, we view the group decision problem as a multi-criteria decision problem, in which individual group members might consider different criteria for evaluating alternatives. Therefore, it is necessary to describe a composite alternative in terms of the union of all individual sets of criteria used by the group members. However, it is unlikely that all partial actions proposed will affect all criteria. Consider, for example, the investment planning problem mentioned earlier. The head of one plant might consider the capacity increase at his plant as one criterion for evaluating investment alternatives. Obviously, investing in equipment at a different plant will not affect that variable. For partial actions, information is therefore only required on those criteria actually affected by that action.

In addition to the data values of partial actions, an *aggregation rule* is needed for every criterion. This aggregation rule describes how the data value of a composite alternative for that criterion is obtained given the data values of the PAs contained in it. In many instances, it is possible to obtain this value by simply adding the individual values of all PAs. Particularly for more qualitative criteria, other aggregation rules like the maximum or minimum operators might also be required.

It is also possible that a composite alternative consists only of partial actions which do not affect some criteria. For example, capacity at plant A will not be changed by an investment program which contains only projects realized in plants B and C. In order to provide a comprehensive evaluation of such composite alternatives, a *status quo* value is needed for each criterion.

The elements mentioned above are necessary components of the problem specification. In addition, criteria values of composite alternatives can be limited by *bounds*. For example, in an investment planning context, it might be required that the total amount invested in a company investment program must not exceed the funds available for investment. Only CAs which are feasible, i.e. which do not violate these bounds, should be passed on to the group for further evaluation.

### 3.2. Generation Process

The generation process for all composite alternatives can be described by a search tree as shown in figure 2. At each level of the search tree, a decision is made about whether one PA should be included in the CA or not. Thus, at each leaf of the tree, a unique CA is generated.

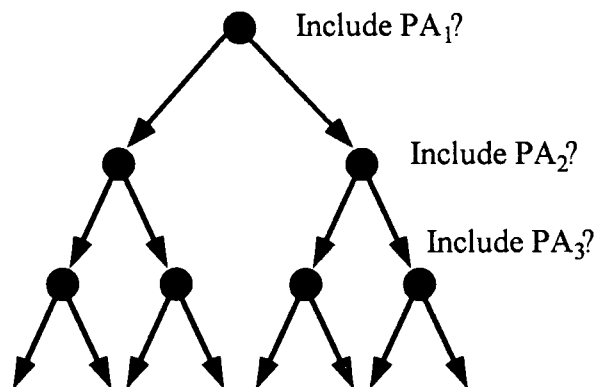


Figure 2: Search Tree for Generating Composite Alternatives

Conceptually, the generation process consists of three phases: firstly, the generation of CAs as sets of PAs. Secondly, the aggregation of data values and finally a testing phase, in which the feasibility of composite alternatives is determined. In terms of the computational efficiency of the generation process, however, this structure is highly inefficient, since it requires the generation of many alternatives, which then turn out to be infeasible. Therefore, a practical implementation technique should try to integrate these phases into one single process.

### 3.3. Implementation Techniques

The structure of the search tree used to generate composite alternatives strongly suggests a recursive algorithm for the generation of the composite alternatives. Since the search process outlined above is very similar to processes used in artificial intelligence, it can be easily implemented using AI languages like Prolog. Figure 3 shows an example of the

generation process for 10 alternatives and 4 criteria written in Prolog. The main predicate of this program is the predicate `generate`. Since Prolog normally uses its built-in recursive process to generate just one solution to the main goal, the program proceeds by generating a solution, storing it in the database and then failing on the main goal, so the backtracking mechanism of Prolog resumes execution and generates the next composite alternative.

The two versions of the predicate `subgen` correspond to the two arcs leaving each node of the search tree. Parameter `N` of these predicates indicates the number of the partial action to be examined, `L` is a list of partial actions already included in the CA and the remaining attributes correspond to the evaluation criteria used in the particular example. All data about partial actions is stored in predicates of the form `project(N, A, K, O, F)`, where `N` is the number of the project and `A, K, O` and `F` contain the data values for the partial actions in all criteria.

The first version of the predicate `subgen` represents the arc of the search tree in which partial action number `N` is not included in the composite alternative. Here, the criteria values `(A, K, O, F)` are simply passed on to the next level of the tree. The second variant of the predicate corresponds to the case where partial action `N` is included in the composite alternative. The data values for partial action `N` are retrieved from the database and aggregation takes place. The example shows that it is easy to use different aggregation mechanisms (like the sum, minimum or maximum operators) in this kind of implementation. It is also possible to incorporate feasibility checks on partially completed composite alternatives at this stage, as shown in the predicate `ok`. Such feasibility checks can also be conveniently written as Prolog predicates. Such predicates can contain more elaborate tests than the one shown here, which only checks the value of one attribute. For example, tests could simultaneously consider the values of several attributes and/or involve more complex algorithms.

The generation process outlined here is a combinatorial problem, which might require excessive computational effort. In the next section, we will therefore explore approaches to reduce this complexity. Such a reduction is necessary both from the point of computational effort during alternative generation, as well as from the point of cognitive burden upon the group members, who have to evaluate the alternatives presented to them.

generate() :-	
subgen(10, L, A, K, O, F),	<i>generate subset</i>
assertz(ca(L, A, K, O, F))	<i>output CA generated</i>
fail.	<i>generate next</i>
subgen(N, L, A, K, O, F) :-	<i>generate subset without project</i>
project(N, _, _, _, _),	<i>find project number N</i>
N1=N-1,	
subgen(N1, L, A, K, O, F).	<i>don't include it</i>
subgen(N, [N L], A, K, O, F) :-	<i>generate including project</i>
project(N, AP, KP, OP, FP),	<i>retrieve data for project</i>
N1=N-1,	
subgen(N1, L, A1, K1, O1, F1),	<i>include it</i>
A=AP+A1,	
K=KP+K1,	<i>aggregation and</i>
min(O, OP, O1),	
max(F, FP, F1),	
ok(A).	<i>feasibility test</i>
subgen(0, , 0, 0, 1000, 0).	<i>empty CA</i>
ok(V) :- V <= 500.	<i>feasibility test</i>

Figure 3: Example for a Generation Process in Prolog

## 4. Reduction of Complexity

### 4.1. Overview

Several strategies are possible to reduce the number of alternatives generated (or considered during the generation process). These strategies can be grouped into three classes, according to the basic concept used:

- Feasibility
- Efficiency
- Filtering

The first two approaches try to avoid the generation of a priori undesirable alternatives. Infeasible alternatives should clearly be avoided. It would also be irrational to consider

alternatives, which are dominated by other alternatives with respect to the criteria considered. The second class of strategies therefore tries to eliminate inefficient alternatives. However, the remaining set of feasible and efficient alternatives might still be large. In this case, filtering techniques have to be applied to generate a subset of alternatives. It should be noted that the use of filtering approaches, in contrast to other approaches, could also eliminate alternatives which are potentially optimal.

## 4.2. Feasibility

As already shown in the Prolog example given above, feasibility of composite alternatives can be checked during the generation process. Furthermore, this test implicitly eliminates an entire section of the search tree. Intuitively, it is possible to eliminate a section of the search tree from consideration, if a bound has been violated in the generation process and the inclusion of any partial action not yet decided upon will not restore feasibility. For example, if an investment program already requires more funds than are available, adding more projects will still result in an infeasible program.

For conditions affecting only a single criterion, infeasibility of a subtree can be checked during the generation process. We call a node in the search tree infeasible, if the aggregate value of all alternatives, already included in the CA at that node, violates its bounds. A subtree is infeasible, if

- a) its root is infeasible and
- b) the aggregate value of the violating criterion increases (or decreases, depending on the type of constraint violated) monotonously with the inclusion of further partial actions.

Part b) of the above condition is met for many aggregation operators under rather weak conditions for the remaining alternatives. For example, if the sum operator is used for aggregation, condition b) is met if all partial actions not yet decided upon have non-negative values in the criterion under consideration.

The above considerations can be formalized as follows. We denote the attributes to be considered in the group decision by  $k \in \{1, \dots, K\}$ . We further assume that there are  $N$  partial actions. The evaluation of PA  $n$  in all attributes is given by the vector  $P_n = (p_{n1}, \dots, p_{nK})$ . For simplicity, we consider only conditions concerning a bound on one attribute. Without loss of generality, we further assume that attribute  $k$  is bounded from above by an upper bound  $b_k$ . We denote the aggregation operator for attribute  $k$  by  $o_k$  if it is used with two arguments and by  $O_k$  if it is used to aggregate across several arguments. At node  $i$  of the search tree, a decision has been made about the partial actions contained in a "Decision Set"  $D_i \subset \{1, \dots, N\}$ , out of which PAs in the "Choice Set"  $C_i \subseteq D_i$  are included in

the composite alternative under construction.  $\overline{D}_i$  is the set of partial actions upon which no decision has yet been made. The evaluation of the composite alternative so far is denoted by  $f_k(i)$ , which is given by:

$$f_k(i) = \bigoplus_{n \in C_i} p_{nk} \quad (1)$$

We define an aggregation operator to be *monotonously increasing* with respect to a set  $S$  of partial actions if

$$z \oplus_k p_{nk} > z \quad \forall n \in S \quad (2)$$

The subtree of the search tree attached to node  $i$  will contain only infeasible nodes if  $f_k(i) > b_k$  and  $\oplus_k$  is monotonously increasing with respect to  $\overline{D}_i$ .

### 4.3. Efficiency

A second approach for reducing the number of composite alternatives generated consists in eliminating all dominated (inefficient) alternatives. In a multi-criteria group decision context, dominance could be defined either in terms of the attributes involved or in terms of group members. In terms of group members, an alternative A dominates an alternative B, if A is considered at least as good as B by all members and strictly better by at least one member. Since we are considering the stage of generating alternatives, this definition, which already requires information about the preferences of members, cannot be applied.

In terms of attributes, alternative A dominates alternative B, if A is at least as good as B in all attributes and strictly better than B in at least one attribute. For the group decision problem considered here, this definition contains two difficulties: firstly the concept of "all" attributes and secondly the concept of being "better" with respect to an attribute.

Group members might consider different sets of attributes in their evaluations. We denote the set used by member  $m$  by  $K_m \subseteq \{1, \dots, K\}$ . "All" attributes could either be interpreted as the original set  $\{1, \dots, K\}$ , which is the union of all individual sets  $K_m$  of attributes or the entire individual set  $K_m$  considered by one member. If we interpret "all" attributes as the union of all individual sets of attributes, it is possible that dominance relations exist with regard to an individual set  $K_m$ , which do not exist at group level. The dominance relation at the group level will therefore eliminate fewer alternatives than an individual dominance relation. The set of alternatives generated will thus contain alternatives which, from the point of view of one member, are dominated. On the other hand, taking into account

individual dominance relations could eliminate alternatives which are not dominated from the point of view of other members.

The second concept which causes difficulties in a group decision context is the concept of being "better" or "at least as good" with respect to one attribute. This is already a statement of preferences which, if we consider dominance at the group level, must be accepted by all group members. If attributes are to be either maximized or minimized, group members must at least agree on the direction of improvement for those attributes. This need not be the case in general, especially in negotiation problems. For example, in a wage bargaining situation, wage increases should be maximized from the union's point of view but minimized from the management's point of view. If group members do not agree on the direction of improvement, dominance relations can only be established between alternatives which have identical values in those criteria. The same holds if single criteria preferences are more complex than simple maximization or minimization, involving, for example, goals to be precisely attained or aspiration levels.

It is also more difficult to incorporate the efficiency criterion into the generation process than the feasibility criterion. At intermediate nodes of the search tree, no decision has yet been made about whether to include the partial actions contained in set  $\bar{D}_i$ . While in some instances, infeasibility of alternatives could be determined independently of the decision about those partial actions, it is less likely that inefficiency can be established for an entire subtree. Partial actions usually lead to an improvement in some attributes, while decreasing the performance in other attributes. If one composite alternative A which does not contain a partial action  $n$  is dominated by another composite alternative B, it might be the case that including PA  $n$  will cause this dominance to disappear if the increase in some attributes due to the inclusion of PA  $n$  is sufficient. This effect is illustrated for the case of two attributes in figure 4.

While the efficiency criterion cannot usually be used to eliminate entire subtrees from the generation process, considering efficiency during generation can still be used to reduce the computational effort. Once a set of alternatives has been established, the subset of efficient alternatives can only be determined by the pairwise comparison of all alternatives, unless elaborate data structures such as quad-trees are used (Habenicht, 1983). During the generation process, while information about the partial actions contained in each CA is readily available, the search for dominating or dominated alternatives can be performed more efficiently using this local information.

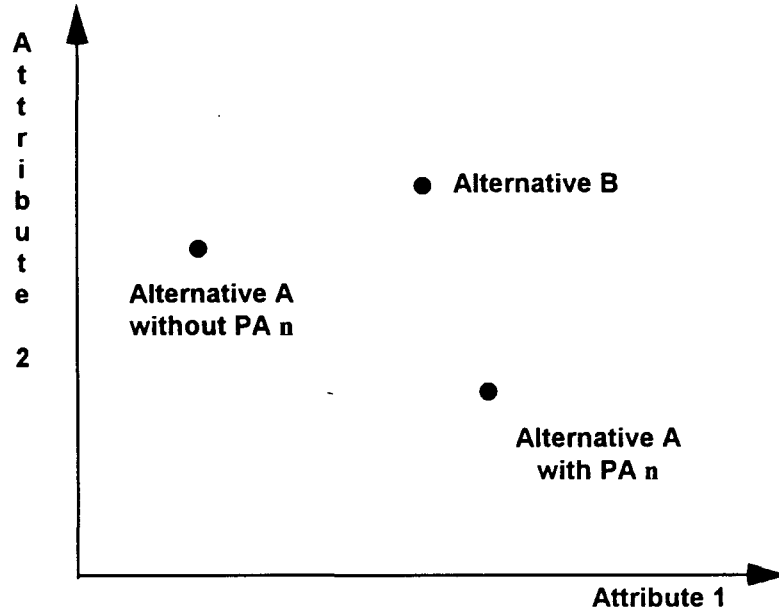


Figure 4: Efficiency of Composite Alternatives

One possibility for exploiting this information is to look for partial actions which can be exchanged and lead to dominance relations. We call partial action  $m$  a *feasible dominating replacement* for partial action  $n$  at node  $i$  if (assuming that all attributes are to be maximized):

$$f_k(i) o_k p_{mk} \geq f_k(i) o_k p_{nk} \quad \forall k \quad n, m \in \bar{D}_i \quad (3)$$

where at least one strict inequality holds and

$$f_k(i) o_k p_{mk} \leq b_k \quad \forall k \quad (4)$$

Possible candidates for feasible dominating replacements can be determined before the generation process by using condition (3). During the generation process, only the feasibility of replacement needs to be checked according to (4), in order to find a possibly dominating CA. Of course, the concept of feasible dominating replacements will only identify some dominance relations among the composite alternatives. A final check using pairwise comparisons must therefore still be made. But since the number of alternatives is reduced in advance by eliminating some dominated alternatives, computation time for this step is reduced.

#### 4.4. Filtering

If the methods outlined above still lead to a large number of feasible and efficient CAs, a further reduction must be achieved by using filtering techniques. In the literature, two approaches for filtering alternatives in multi-attribute problems were proposed: filtering by thresholds and random filtering.



The basic idea of filtering by threshold levels (Abonyi, 1983), also called "conjunctive method" (Hwang/Yoon, 1981) is to eliminate undesirable alternatives which do not reach certain attribute values. However, these threshold levels already introduce some preference information at a stage before the actual evaluation of alternatives is performed. This might be problematic, especially in a group decision context, in which individual evaluations (on which the threshold levels must be based) might change during the process, due to influence from other group members.

In random filtering (Steuer, 1986), one tries to create a statistical sample of the efficient surface by randomly selecting widely dispersed alternatives. This approach avoids the use of a priori preference information and therefore seems to be more appropriate in a group decision context.

In multi-objective programming problems, a distinction is often made between filtering in decision space, where equally spaced values of the decision variables are used and in criterion space, where one tries to generate an evenly dispersed set of objective values (Steuer/Harris, 1980). A similar distinction can be made in the case of composite alternatives, where the composition of CAs takes the position of decision variables. Taking this idea one step further, one could consider the attribute values of a CA as the result of an experiment with the selection of PAs as control variables, and use techniques from experimental design, like latin squares or specific fractioning techniques for two-level factors (John, 1971) to obtain an evenly spread sample of CAs. These techniques can be directly incorporated into the generation process, while for random sampling techniques, the entire set of alternatives must first be generated. However, in the methods from experimental design, the amount of data points generated is determined by the method applied, while in random sampling techniques in attribute space, the sample size can be directly controlled. Random sampling methods are therefore more flexible.

## 5. Conclusions

In this paper, we have introduced the concept of composite alternatives, which form an important aspect of many actual group decision problems. We have studied the problem of generating a set of composite alternatives from partial actions and methods for reducing the number of alternatives contained in that set.

The concept of composite alternatives developed in this paper still requires further research, both at the conceptual level and the level of implementation. Conceptually, the advantages and disadvantages of generating discrete alternatives vs. multiobjective programming approaches need to be analyzed in more detail. Systems based on both approaches have

been proposed in the literature, comparative (possibly experimental) studies of such systems could help to resolve that question.

The computational techniques developed in this paper can also be expanded upon. The basic generation process presented in section three, as well as its extensions concerning the generation of feasible alternatives only, have already been incorporated into an experimental system (Vetschera, 1991b). Filtering methods are also available in the form of computer codes (Steuer, 1986), but they have not yet been tested in the context described here. No computational experiments have been performed on the incorporation of efficiency tests into the generation process and on the reduction in computational effort which can be achieved by this technique. These topics create a broad set of alternatives for future research.

## References

- Abonyi, G. (1983): *Filtering: an approach to generating the information base for collective choice*. Management Science 29: 409-418.
- Bronisz, P.; Krus, L.; Wierzbicki, A. P. (1989): *Towards Interactive Solutions in a Bargaining Problem*. In: A. Lewandowski and A. P. Wierzbicki (Eds.): *Aspiration Based Decision Support Systems*. Springer, Berlin et al.: 251-268.
- Bui, T. X. (1985): *N.A.I.: A consensus seeking algorithm for group decision support systems*. Proceedings, 1985 International Conference on Cybernetics and Society, Tucson, Arizona: 380-384.
- Bui, T. X. (1987): *Co-oP - A Group Decision Support System for Cooperative Multiple Criteria Group Decision Making*. Springer, Berlin et al.
- Habenicht, W. (1983): *Quad Trees, A Datastructure for Discrete Vector Optimization Problems*. In: P. Hansen (Ed.): *Essays and Surveys on Multiple Criteria Decision Making*. Springer, Berlin et al.: 136-145.
- Hwang, C.-L.; Yoon, K. (1981): *Multiple Attribute Decision Making - Methods and Applications. A State of the Art Survey*. Springer, Berlin et al.
- Isermann, H. (1984): *Investment and Financial Planning in a General Partnership*. In: M. Grauer and A. P. Wierzbicki (Ed.): *Interactive Decision Analysis*. Proceedings, Laxenburg 1983. Springer, Berlin et al.: 175-185.
- John, P. W. M. (1971): *Statistical Design and Analysis of Experiments*. Macmillan, New York.
- Kersten, G. E.; Michalowski, W.; Szpakowicz, S.; Koperczak, Z. (1991): *Restructurable Representations of Negotiation*. Management Science 37: 1269-1290.
- Lewandowski, A.; Johnson, S.; Wierzbicki, A. P. (1987): *A Prototype Selection Committee Decision Analysis and Support System. SCDAS: Theoretical Background and Computer Implementation*. In: Y. Sawaragi, K. Inoue and H. Nakayama (Eds.): *Toward Interactive and Intelligent Decision Support Systems. Volume 2*. Springer, Berlin et al.: 358-365.
- Matwin, S.; Szpakowicz, S.; Koperczak, Z.; Kersten, G.; Michalowski, W. (1989): *Negoplan: An Expert System Shell for Negotiation Support*. IEEE Expert 50-62.
- Roy, B. (1991): *The Outranking Approach and the Foundations of ELECTRE Methods*. Theory and Decision 31: 49-73.
- Saaty, T. L. (1987): *Resolution of Retributive Conflicts*. In: G. K. Rand (Ed.): *Operational Research '87*. Elsevier Science, Amsterdam: 549-565.
- Steuer, R. E. (1986): *Multiple Criteria Optimization*. J. Wiley & Sons, New York.
- Steuer, R. E.; Harris, F. W. (1980): *Intra-Set Point Generation and Filtering in Decision and Criterion Space*. Computers and Operations Research 7: 41-53.

- Vetschera, R. (1990): *Group Decision and Negotiation Support - A Methodological Survey*. OR Spektrum 12: 67-77.
- Vetschera, R. (1991a): *Entscheidungsunterstützende Systeme für Gruppen*. Physica-Schriften zur Betriebswirtschaft. Physica, Heidelberg.
- Vetschera, R. (1991b): *GDSS-X: An Experimental Group Decision Support System for Program Planning*. In: P. Korhonen, A. Lewandowski and J. Wallenius (Eds.): *Multiple Criteria Decision Support*. Springer, Berlin: 369-376.