Distillation Absorption 2010
A.B. de Haan, H. Kooijman and A. Górak (Editors)

A MEMETIC ALGORITHM FOR THE DESIGN OPTIMIZATION OF REACTIVE DISTILLATION COLUMNS

Maren Urselmann, Sebastian Engell
Process Dynamics and Operations Group, Technische Universität Dortmund, 44221 Dortmund, Germany, {maren.urselmann|sebastian.engell }@bci.tu-dortmund.de

Abstract
This contribution deals with the optimization of the design of reactive distillation columns by using a memetic algorithm (MA) which is a combination of an evolution strategy (ES) and a mathematical programming (MP) solver. The standard approach to solve such problems is to formulate them as large MINLPs but then the computational effort needed for the solution process grows substantially if the number of discrete variables increases, e.g. if a restriction on the number of feeds is introduced. Another problem is that the nonlinear solvers only provide a single local optimum. The MA overcomes this problem by addressing the optimization of the discrete and the global search in the space of the continuous design variables, while continuous sub-problems are efficiently solved by an MP solver. A comparison of the results of this new approach with results of commercial MINLP techniques shows that the MA can efficiently handle the global design optimization problem at hand and that it is the only algorithm that found the global solution if the number of discrete variables increases.

Keywords: Conceptual design, mathematical optimization, memetic algorithm, reactive distillation, global optimization

1. Introduction

A current trend in process design is towards integrated processes, i.e. the integration of reactive and separating functionalities into a single apparatus as, e.g., a reactive distillation column. Compared to the classical serial arrangement of unit operations, this advanced concept has the potential to decrease the dimensions of the equipment and to increase the degree of heat integration. Furthermore, it provides the opportunity to overcome chemical and thermodynamical boundaries, such as chemical equilibria or distillation boundaries due to azeotropes. Separations of non-ideal mixtures with simultaneous chemical reactions belong to the most difficult design problems and should be solved in an integrated fashion. The design of a reactive distillation column constitutes a constrained mixed combinatorial optimization problem which is amenable to MINLP techniques. In practice, such problems are often hard to solve due to non-linear and integrality constraints and the nonconvexity of the continuous sub-problems.

In our previous work[1,2], the optimization-based design of reactive distillation columns was addressed by using MINLP techniques. The solution procedure is based on a decomposition of the MINLP problem into an IP-master-problem (optimization of the number of trays and of the location of the feed streams) and NLP-sub-problems (optimization of continuous variables for fixed discrete variables). In order to reduce the complexity of the problem, the number of feed trays was fixed to two. The nonconvex continuous sub-problems were tackled by a combination of the scatter-search based multi-start algorithm OQNLP and the local NLP solver CONOPT. The IP-master-problem was addressed by the branch and bound solver SBB. The application of decomposition based procedures lead to better performances than approaches without decomposition[3].

An extension of the model by a variable number of feeds that may lead to better results than fixing the number of feed trays to only two substantially increases the computational effort that is needed for the solution. In case of the decomposition based approach described above, the complexity of the IP-master-problem and the number of NLP-sub-problems increase depending on the maximum number of feed streams. These models are too complex to be solved to global optimality by the use of multi-start algorithms in reasonable time[1]. Recently[4], a memetic algorithm (MA) for the global solution of reactive distillation problems without restrictions on the number of feeds was introduced. By the use of this method, the computational effort needed for a local search of the continuous design optimization

without restrictions on the number of feeds and for a fixed number of trays could be reduced by 75% in comparison to the reference algorithm (OQNLP/CONOPT). The MA consists of an evolutionary algorithm (EA) and the mathematical NLP solver CONOPT. The EA generates initial points for the local solver. It works in the space of the design variables whereas the state variables of the column designs are computed by the same solver that performs the local optimization. This concept was successfully extended by a restriction on the number of feed streams for each feed[5].

In this work, both variants of the MA, i.e. the approach with restrictions on the number of feeds ($MA_{MINLP-CF}$) and without (MA_{MINLP}), are extended to the optimization of the number of trays. In order to exploit the ability of EAs to handle integrality constraints and discontinuous cost functions and the ability of mathematical programming (MP) methods to efficiently solve large continuous problems locally, the EA addresses the optimization of the design variables of the problem, including all discrete variables, i.e. the number of trays and the number and the location of the feed streams, whereas the NLP solver is used to solve the continuous sub-problems which arise by fixing all discrete variables. The results of the MAs are compared to the results of the commercial solvers OQNLP/CONOPT, SBB/CONOPT and SBB/OQNLP/CONOPT that serve as references. The reference algorithms optimize the model of the superstructure in which the number of trays and the location of the feed streams are modeled by binary variables. Design problems of reactive distillation columns have also been addressed by stochastic algorithms, but applications of evolutionary algorithms are very few, e.g.[6].

2. The case study

As a case study, the optimization-based design of a reactive distillation column for the production of MTBE from isobutene and methanol (*IB + MeOH ↔ MTBE*) in the presence of n-butane at a pressure of 8 bar is considered. The reaction is kinetically controlled, equilibrium limited and heterogeneously catalyzed. The substance system exhibits 3 binary azeotropes: MeOH/MTBE, MeOH/IB and MeOH/butane. The desired purity of the product is 99 mole-%. The total amount of the feed streams is fixed ($F_{1,tot}$ *= 6.375 mole/s MeOH,* $F_{2,tot}$ *= 8.625 mole/s IB/n-butane*). Structural and operational parameters - e.g. the number of trays and the reflux ratio - have to be determined such that the annual profit of the column is maximized.

The model of the tray column is based on the MESH equations (material balance, equilibrium condition, summation condition, enthalpy balance) which are extended by reaction terms[3]. The non-ideality of the behavior of the mixture is described by activity coefficients that are calculated by Wilson's approach. All thermodynamic parameters are taken from[7]. The reaction kinetics are based on activities, the temperature dependency of the kinetic constants is given by an extended Arrhenius approach and the temperature dependency of the vapor pressure is modeled by the Antoine equation. A Murphree stage efficiency for the vapor phase reflects the non-ideality of the separation functionality. The superstructure of the process comprises $N = 60$ trays of which only a subset may be included in the optimal solution. The inclusion of a certain tray k is coded by a binary activation variable φ_k (1 indicates activity and 0 indicates inactivity). The reboiler and the condenser are modeled as trays without reaction. Inactive trays are moved to the top of the column. The objective is to maximize the annual profit which is calculated by the annual revenues for the products minus the annualized investment cost, annual operating cost and annual cost for raw materials. The investment cost are calculated by heuristic functions for the cost of the column shell, the internals, the catalyst, the condenser and the reboiler; the operating cost is calculated by the heat loads for heating and cooling.

The set of design variables consists of the amounts of both feeds $i = 1, 2$ on the trays $k = 1, ..., N$ denoted by $F_i(k)$, the amounts of catalyst on the trays $k = 2, ...,N-1$ denoted by $E_{cat}(k)$, two variables α_{top} and $\alpha_{bottom} \in (0, 1)$ for the reflux ratio at the top and the ratio of the evaporation rate to the product removal at the bottom of the column and the binary activation variables φ_k for the trays $k = 2, \ldots, N - 1$. The models consist of a large number of algebraic equations formulated in the modeling language GAMS. Different models are used for the different algorithms ($MTBE_{MINLP}$, $MTBE_{MINLP-CF}$, $MTBE_{NLP}$, and $MTBE_{Sim}$). The sizes of the models are given in Table 1.

2.1 $MTBE_{MINLP}$

$MTBE_{MINLP}$ is the model of the superstructure of the MTBE column as described above without considering restrictions on the number of feeds. It is assumed that fractions of both feed streams can enter the column on each tray including the reboiler and the condenser. Except of the $N - 2 = 58$

binary activation variables φ_k all variables of the model are continuous. For a more detailed description of the model see[1,2].

2.2 MTBE$_{MINLP-CF}$

MTBE$_{MINLP-CF}$ is the extension of the model MTBE$_{MINLP}$ by a restriction on the number of feed streams. For each feed *i = 1, 2* and each tray of the column *k = 1, …, N* a binary variable $\delta_{i,k}$ indicates if a fraction of feed stream *i* enters the column on tray *k* ($\delta_{i,k} = 1$) or not ($\delta_{i,k} = 0$). A maximal number of three feed streams nF_i^{max} for each feed *i* is imposed by the following inequality:

$$\sum_{k=1}^{N} \delta_{i,k} \leq nF_i^{\max} \quad \forall i \tag{1}$$

Additional constraints in the model ensure, that there can only be feeds on active trays of the column and that there is only feed *i* on tray *k* if $\delta_{i,k}$ equals 1. By restricting the number of feeds per stream to a maximum of three the number of discrete variables increases to *3N – 2 = 178*.

2.3 MTBE$_{NLP}$

MTBE$_{NLP}$ is the model of the continuous sub-problems which arise by fixing all discrete variables of MTBE$_{MINLP}$. The maximal number of trays N is fixed to a value between 10 and 60 and all of these trays are active. Hence, the size of the model depends on *N*.

2.4 MTBE$_{Sim}$

MTBE$_{Sim}$, also denoted as the simulation model, is the model used to determine the values of the model variables that correspond to a certain column design. It comprises a subset of the equations and of the inequalities of the optimization model MTBE$_{NLP}$. The design variables here are removed from the set of free variables, and the equations and the inequalities that restrict the feasible values of the design variables are removed from the set of constraints as well.

Table 1. Model sizes

	MTBE$_{MINLP}$	MTBE$_{MINLP-CF}$	MTBE$_{NLP}$	MTBE$_{Sim}$
binary variables	58	178	-	-
continuous variables	9134	9134	152N + 14	149N + 4
constraints	9430	9552	155N + 14	149N + 4

3. The memetic algorithm

Memetic algorithms[4,5,8] are hybrid evolutionary algorithms coupled with local refinement strategies. In this work, an evolution strategy[9] (ES) which is a special variant of an EA is used. ES are designed for continuous search spaces and have a special feature: the strategy parameters, e.g. the parameters to determine the mutation strength, are adapted during the search (self-adaptation).

3.1 Structure of the memetic algorithm

The structure of the memetic algorithm used here is depicted in Figure 1. The optimization procedure starts with a feasible random initialization of the first population. In order to evaluate the μ individuals of the population, the corresponding model variables are computed by CONOPT by solving the simulation model MTBE$_{Sim}$. The resulting point in the space of all variables represents a possible column design which is used as a starting point for the local optimization in the space of all continuous variables. This local search is also performed by CONOPT based upon the model MTBE$_{NLP}$. According to the evolutionary model of Lamarck, the genes of the individuals are replaced by the values of the design variables of the corresponding local optimum. As long as

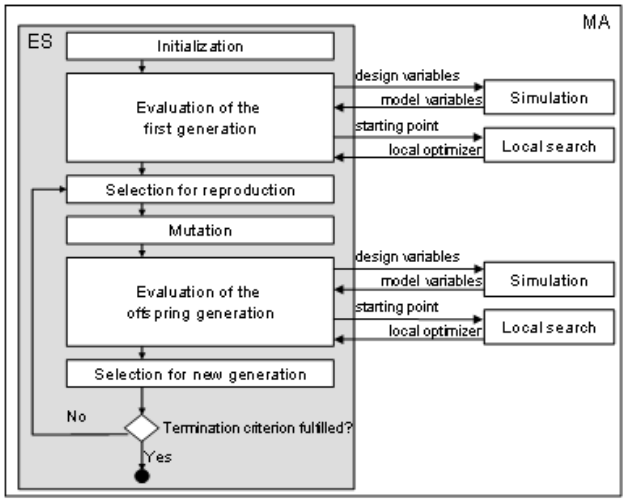

Figure 1. Structure of the memetic algorithm

no feasible column design with *N* trays is found, all model variables within the simulation model are initialized with the value 1. During the subsequent optimization, the values of the model variables of the nearest feasible point found so far (measured by the Euclidean norm) are used as initial values. In previous work[4] it could be shown that the use of this initialization procedure can significantly reduce the computational effort that is needed for the simulation. The generation cycle of the ES starts with a random selection of λ individuals for the reproduction. These individuals are mutated as described in Section 3.3 and become offspring individuals which are evaluated in the same manner as the individuals of the initial population. Then the population for the next generation cycle is selected by choosing the μ best individuals out of the set of offspring and parent individuals which do not exceed a maximal 'life-span' of κ generations. This selection procedure, called the (μ, κ, λ)–selection, is typically used in ES. The generation cycle stops if a predefined termination criterion is fulfilled e.g. a time limit or a generation limit.

3.2 Representation of the individuals

In evolutionary strategies, individuals are represented by a vector that represents the object variables of the optimization problem (here: the design variables) and a strategy parameter vector. In case of the optimization without restrictions on the number of feeds (MA_{MINLP}) the genes of an individual are given by the integer variable *N* that represents the number of trays and by the continuous design variables described in Section 2. In order to avoid the existence of redundant variables and to reduce the size of the model of the continuous sub-problems, a representation is chosen that allows individuals of different sizes (depending on the number of trays *N*) to be members of the same population. Thus the binary activation variables $\varphi_{i,k}$ are not required. The strategy parameter vector contains one step size parameter for the mutation of the number of trays *N*, one for the amounts of feeds for each feed *i*, *N* - 2 parameters for the amount of catalyst on the trays and two for the parameters α_{top} and α_{bottom}.

In case of the formulation $MA_{MINLP-CF}$, the vector of object variables is extended by the discrete variables noF_i that represent the number of feed streams per feed and by two vectors $indF_i$ with noF_i distinct discrete elements $g \in \{1, \ldots, N\}$ that represent the indices of the feed trays for feed *i*. The extended strategy parameter vector contains four additional parameters: one parameter for each variable noF_i and one for each vector $indF_i$. The maximal number of feeds per stream is three.

3.3 Initialization

The size of the object variable vector depends on the number of trays *N*. Therefore, the number of trays is chosen first which is done randomly with a uniform distribution within the range of *10* to *60*. In case of the formulation (MA_{MINLP}), the continuous design variables are chosen next. This is done randomly within their bounds and respecting the constraints defined on the design variables, i.e. the fixed sum of the total amount of feed *i*. In case of the formulation ($MA_{MINLP-CF}$), the number of feed streams for each feed (noF_i) and the noF_i locations of these streams have to be chosen before the continuous variables can be initialized. After the choice of the feed locations, the amounts of feeds on all other trays are fixed to zero. These variables are also fixed in the model $MTBE_{NLP}$ which is used by the local solver. For more information see[5].

3.4 Mutation

In evolution strategies, the mutations of the strategy parameters and of the object parameters are done consecutively. The strategy parameters are mutated first. In this work, reflection is used to avoid that a parameter exceeds its feasible domain. Because of the variable-length representation of the individuals, the mutation is done in a hierarchical fashion. First, the strategy parameter of the number of trays is mutated by the standard mutation operator for continuous variables[9] followed by the mutation of *N* which is done by the use of the mutation operator for integer search spaces[10], here denoted as mut_I. If the number of trays does not change during the mutation procedure (i.e. $mut_I(N) = N$) the mutation operator developed for a fixed number of trays[4,5] is applied. In case of a change in the number of trays ($mut_I(N) \neq N$), the size of the offspring differs from the size of the parent individual.

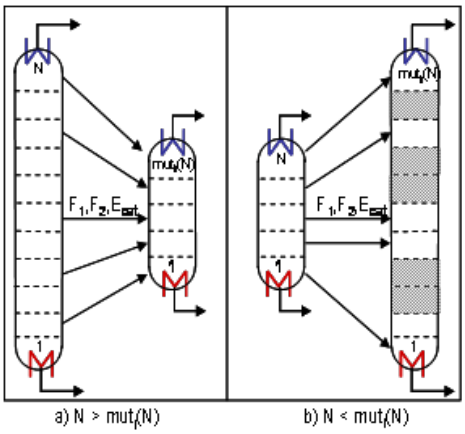

Figure 2. Tray-by-tray assignment of the variables F_1, F_2 and E_{cat}

The mutation takes place by assigning the F_1, F_2 and E_{cat} variables from trays of the parent individual to trays of the offspring. In Figure 2, the tray-by-tray mutation procedure of the object variables is illustrated.

If N is greater than $mut_l(N)$ (see Figure 2a), $mut_l(N)$ distinct tray indices $j_n \in \{1, \ldots, N\}$ with $n = 1, \ldots, mut_l(N)$ are chosen randomly. Then they are sorted and the variables $F_1(j_n)$, $F_2(j_n)$ and $E_{cat}(j_n)$ are assigned to the trays $k = 1, \ldots, mut(N)$ of the offspring individual, successively. If N is less than $mut_l(N)$ (see Figure 2b), N distinct tray indices $j_n \in \{1, \ldots, mut(N)\}$ are chosen and sorted and the corresponding variables of the parent individual are assigned to the trays of the offspring. In the latter case, some of the offspring variables - those which correspond to the gray trays in Figure 2b) - are not assigned after this procedure. These variables are initialized randomly within their bounds. In case of the formulation $MA_{MINLP-CF}$, the next step is the determination of the number of the feeds and of the indices of the feed trays of the offspring to assign the values of the discrete variables. Repair procedures are applied if the number of feed streams of a feed i is zero and/or if the total amount of feed i is not equal to the predefined value. The values of the object variables α_{top} and α_{bottom} are inherited without changes as well as the strategy parameters with the exception of the parameter for the number of trays N.

4. Reference algorithms

Two commercial algorithms were used before with some success to solve the case study at hand[1,2]. These two algorithms serve as a reference for the performance of the new approach.

4.1 OQNLP/CONOPT

OQNLP is a scatter search based multi-start heuristic that generates different starting points for a local NLP solver in the space of all variables. Each candidate point has to pass two different filters (a merit filter and a distance filter) to be accepted as a starting point for the local solver CONOPT. OQNLP was designed for the global optimization of smooth constrained nonlinear programs (NLPs) but can also handle discrete variables. In this case, OQNLP generates starting points in the space of all variables, fixes the discrete variables and calls CONOPT for the local search of the continuous sub-problem.

4.2 SBB/CONOPT and SBB/OQNLP/CONOPT

SBB is based on a combination of the standard branch and bound (B&B) method known from MILP problems and a NLP solver. The integrality requirements are totally relaxed in the root node and reinforced for one variable after the other and from layer to layer. At each node of the search tree, a lower bound on the optimal solution is generated by solving an NLP (integer-relaxed MINLP). This bound is only a valid lower bound on the optimal solution if the solution found by the NLP solver is the global optimum of the continuous sub-problem represented by the node. This can only be guaranteed in case of a convex problem otherwise it could be possible that the global solution is excluded from the search by SBB. The solution quality of SBB can be improved[1] by solving the continuous sub-problems more than once starting the NLP solver from different starting points. Therefore, in addition to the use of SBB with CONOPT, SBB is also used with the multi-start algorithm OQNLP in combination with CONOPT. Solving the sub-problems more than once in each node leads to a significant increase in the computational effort needed for the solution, depending on the number of calls of CONOPT per node.

5. Results

All algorithms were tested on a PC with 3.06GHz and 2GB RAM. Algorithms with stochastic influences, i.e. MA_{MINLP}, $MA_{MINLP-CF}$, OQNLP/CONOPT and SBB/OQNLP/CONOPT, were tested 5 times and the median performances of these runs are compared with the deterministic runs of SBB/CONOPT. For a fair comparison of all algorithms, a parameter tuning was done in preliminary test runs. The strategy parameters $\mu = 5$, $\kappa = 5$, $\lambda = 10$ for MA_{MINLP} and $\mu = 7$, $\kappa = 5$, $\lambda = 14$ for $MA_{MINLP-CF}$ lead to the best results. The termination criterion is a limit of 25 generations in case of the MA. OQNLP/CONOPT was stopped after 4 hours and the termination criterion for SBB was a limit of 10,000 nodes.

In both cases, with and without restrictions on the number of feed streams, the column design shown in Figure 3a) is the best design known so far. It yields an annual profit of 965,592 €. Without restrictions on the number of feed streams, this solution was found by the MA in all test runs. The median time needed to find this solution was 50 min and 40 sec. None of the other algorithms was

able to find this column design in reasonable time. SBB/CONOPT was the fastest algorithm. It terminated regularly without reaching the predefined node limit after 7 min and 35 sec. The best solution was found after 5 min and 31 sec and has an annual profit of 964,721 € which is slightly worse than the best solution found by the MA. The two solutions are structurally different. The regular termination of SBB/CONOPT shows that the global solution was excluded from the search because of the non-convexity of the sub-problems in the nodes. In the application of SBB/OQNLP/CONOPT, 6 CONOPT calls per node were necessary to find the best solution known so far within 3 of 5 test runs. The median time that is needed to find this solution is 470 min and 23 sec which is approx. 9 times the time needed by the MA.

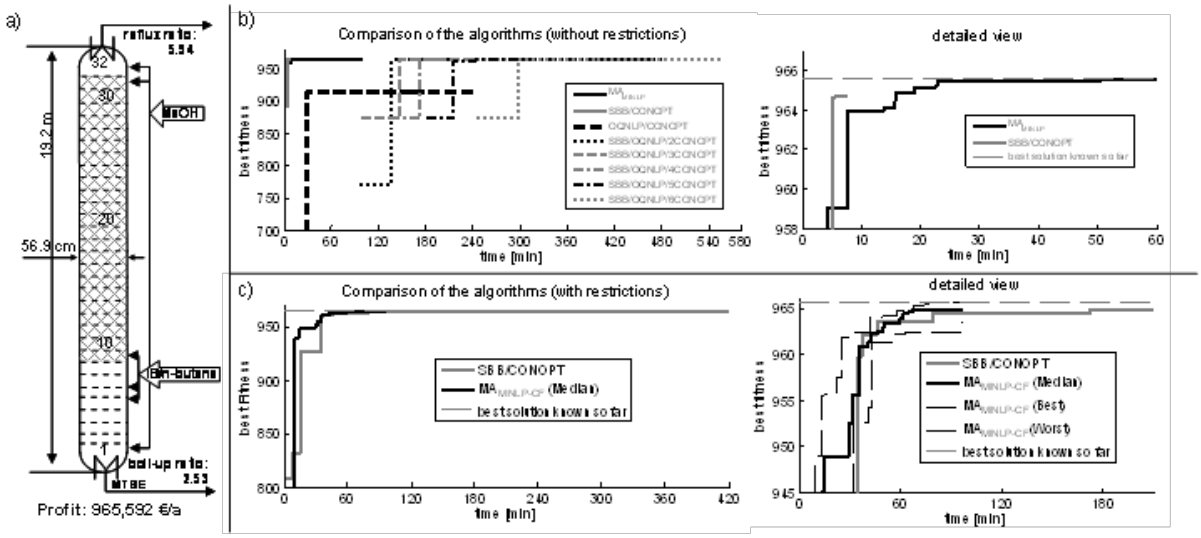

Figure 3. a) best column design known so far, b) progress curves of the algorithms (without restrictions), c) progress curves of the algorithms (with restrictions on the number of feeds)

In case of the formulation with restrictions on the feeds, the MA found the best solution in 1 of 5 test runs. The median profit of the best solution found by the MA was 964,775€. The median time to find this solution was 67 min and 20 sec. This solution was also found by SBB/CONOPT after 171 min and 53 sec. SBB/CONOPT terminated after 6 hours because the given node limit was reached. OQNLP/CONOPT could not find any feasible solution within the given time limit. SBB/OQNLP/CONOPT with more than one CONOPT call did not terminate within several days.

6. Conclusions & Outlook

A memetic algorithm that consists of an evolution strategy and a local NLP solver was successfully applied to the global design optimization of a reactive distillation column with and without restrictions on the number of feeds. In case of the formulation without restrictions, the MA found the best solution known so far with a success rate of 100%. The MINLP solver SBB/CONOPT could find a solution with approximately the same quality faster, but the global optimum was excluded from the search. Multiple starts of the local solver in each node solve this problem, but then the computational effort increases by about two orders of magnitude. In case of the formulation with restrictions on the number of feeds, the MA was the only algorithm that found the best solution known so far. In future work, the MA will be applied to the design optimization of the MTBE column with an optional external reactor.

References

1. S. Barkmann, G. Sand, S. Engell, *Chemie Ingenieur Technik,* 80(2008) 107–117.
2. S. Barkmann, G. Sand, S. Engell, *Proc. ECCE*, 6(2007) 435-436.
3. J. Jackson, I. Grossmann, *Comp. and Chem. Eng.,* 25(2001) 1661-1673.
4. M. Urselmann, G. Sand, S. Engell, *Proc. IEEE CEC,* (2009) 1721-1728.
5. M. Urselmann, S. Engell, *ESCAPE 20*, (2010), Accepted for publication.
6. B.V. Babu, M. Khan, *Asia-Pacific Journal of Chemical Engineering*, 2(2007), 322-335.
7. B. Beßling, Dr.-Ing.-Dissertation, Fachbereich Chemietechnik, Universität Dortmund, 1998.
8. P. Moscato, *Caltech Concurrent Computation Program*, report 826, (1989).
9. H. G. Beyer, H. P. Schwefel, *Natural Computing: An International Journal* 1(1) (2002), pp. 3-52.
10. G. Rudolph, *Proc. PPSN III, LNCS Nox*, Springer, Berlin (1994), pp. 193-197.