TECHNISCHE UNIVERSITÄT DORTMUND

REIHE COMPUTATIONAL INTELLIGENCE

COLLABORATIVE RESEARCH CENTER 531

Design and Management of Complex Technical Processes and Systems by means of Computational Intelligence Methods

Analysis of a memetic algorithm for global optimization in chemical process synthesis

Maren Urselmann, Guido Sand and Sebastian Engell

No. CI-254/08

Technical Report ISSN 1433-3325 December 2008

Secretary of the SFB 531 · Technische Universität Dortmund · Dept. of Computer Science/LS 2 · 44221 Dortmund · Germany

This work is a product of the Collaborative Research Center 531, "Computational Intelligence," at the Technische Universität Dortmund and was printed with financial support of the Deutsche Forschungsgemeinschaft.

Analysis of a memetic algorithm for global optimization in chemical process synthesis

M. Urselmann, G. Sand and S. Engell, *Process Dynamics and Operations Group, TU Dortmund*

***Abstract*— Engineering optimization often deals with very large search spaces which are highly constrained by nonlinear equations that couple the continuous variables. In this contribution the development of a memetic algorithm (MA) for global optimization in the solution of a problem in the chemical process engineering domain is described. The combination of an evolutionary strategy and a local solver based on the general reduced gradient method enables the exploitation of a significant reduction in the search space and of the ability of local mathematical programming solvers to efficiently handle large continuous problems containing equality constraints. The global performance of the MA is improved by the exclusion of regions that are defined by approximations of the basins of attraction of the local optima. The MA is compared to the combination of a scatter search based multi-start heuristic using OQNLP and the local solver CONOPT.**

I. Introduction

The optimal design of a reactive distillation column with respect to an economic function is an example of real-world problems in the chemical process engineering domain. Reactive distillation columns are chemical plants that combine chemical reaction with distillation (the separation of liquid mixtures based on differences in their volatilities) within a single processing unit [1]. The aim of the optimization procedure is to find the optimal column design out of all suitable alternatives with respect to the total annualized cost.

The standard approach to handle such problems is to formulate them as mixed-integer nonlinear programming (MINLP) problems and to solve them by mathematical programming methods. Such an MINLP can be stated as follows:

$$\begin{aligned} min\, F \quad &= f(x,y) && (1)\\ st. \quad h(x,y) \quad &= 0 && (2)\\ g(x,y) \quad &\leq 0 && (3)\\ x \in X, \quad & y \in N, \end{aligned}$$

where F is the cost function, $h(x, y) = 0$ are the equations that describe the behavior of the system (mass and heat balances, geometry, etc.), and $g(x, y) \leq 0$ are the inequalities that define the specifications or constraints for feasible choices [2]. The continuous variables x denote the state variables (in this paper denoted by the term *model variables*) and the *design variables* such as the feed flows or the amounts of catalyst on each tray of a distillation comlumn. The integer variables y correspond to design variables, e.g. the number of trays of a column.

By fixing the integer variables the problem can be divided into several continuous subproblems (NLPs) of different dimensions depending on the number of trays. These subproblems have dimensions of $10^3 - 10^4$ continuous variables, equations and constraints and include nonlinear equations that restrict the continuous variables, what makes them very difficult to solve.

For the solution of these optimization problems, local mathematical programming (MP) methods are available [3], that can efficiently handle large problems and have been applied - with some success - to reactive distillation processes before ([4], [5]). Because of the non-convexity of the problems, only local optima can be found by these methods.

It is possible to reduce the search space of the continuous optimization problem to the space of the design variables (in the following stated as $d \in D$), which comprises about 2% of all variables. In this case, a computationally expensive simulation is needed to determine the corresponding state variables in order to evaluate the cost function and the constraints. The computation time needed for a single simulation is nearly half of the time that an NLP solver needs for a local search in the space of all variables.

In this contribution, the development, the application and the analysis of a memetic algorithm for the global optimization of a process design with fixed integer variables are described.

Since evolutionary algorithms are devised to escape from local optima and in order to exploit the ability of the MP methods to efficiently solve large continuous problems locally, a memetic algorithm was developed that combines an evolutionary strategy (ES) [6] with a local NLP-solver based on the general reduced gradient method (CONOPT [7]). Within the memetic algorithm, the ES addresses the optimization of the design variables and every individual of the ES is improved by a local search of the MP solver. Thus every feasible individual within the population of the ES represents a local optimum of the problem at hand.

An ideal search strategy would utilize variation operators that change the genes of the parent individuals in such a way that every offspring is in the basin of attraction of a different local optimum. In order to guide the search to different local optima, regions are defined that approximate the basins of attraction of each local optimum. The definition of the regions is based on information on the starting point of the local optimization and the resulting local optimum.

M. Urselmann and S. Engell are with the Department of Biochemical- and Chemical Engineering, Technische Universität Dortmund, Emil-Figge-Str. 72, 44227 Dortmund (phone: 0049-231-755-7378; fax: 0049-231-755-5129; email: maren.urselmann@bci.tu-dortmund.de). G. Sand is with ABB Corporate Research, Wallstadter Str. 59, 68526 Ladenburg, (phone: 0049-6203-716354, fax: 0049-6203-716253, email: guido.sand@de.abb.com).

In the sequel these regions are called *tabu zones*. After its definition, a tabu zone is excluded from the subsequent search procedure.

In Section II the case study considered here is introduced. Then the structure of the memetic algorithm and the approximation of the basins of attraction is explained in Section III. The main results of the numerical tests and the analysis of the MA are stated in Section IV. Finally, Section V comprises the conclusion and the outlook.

II. The case study

The case study considered here is the optimal design of a reactive distillation column for the production of the anti-knock product methyl tert-butyl ether (MTBE). In Figure 1 the schematic structure of a reactive distillation column is shown. MTBE is synthesized from two feed streams with a purity of at least 99 mole-%. Feed stream one (F_1) is composed of $F_1^{sum} = 6.375$ mole/s of methanol (MeOH); feed stream two (F_2) comprises $F_2^{sum} = 8.625$ mole/s of a mixture of isobutene (IB) and butane. The number of trays, denoted by N, is fixed. The reboiler and the condenser of the distillation column are modeled as trays and they are denoted by tray numbers $k = 1$ and $k = N$. It is assumed that fractions of both feed streams can enter the column on each tray of the column including reboiler and condenser. The chemical reaction is catalyzed heterogeneously. It is assumed that there can be a certain amount of catalyst (E_{cat}) on each tray of the column, restricted by the volume of a tray. No reaction is taking place in the reboiler nor in the condenser. The diameter of the column is denoted by Dia.

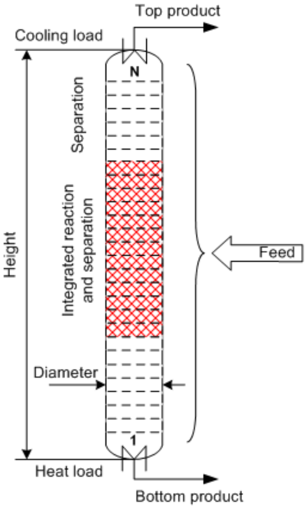

Fig. 1. Structure of a reactive distillation column

The objective (1) is to minimize the annual cost which is calculated by the annualized investment cost, annual operating cost and annual cost for raw materials minus annual revenues for the products. The investment cost are calculated by heuristic functions for the column shell, the internals, the catalyst, the condenser and the reboiler; the operating cost are calculated by the heat loads for heating and cooling.

The model variables are related to the operating condition inside the column, e.g. the pressure, the steam velocities or the concentrations of the substances.

The set of design variables $d \in D$ consists of the amounts of both feeds $i = 1, 2$ on each tray $k = 1, ...N$ denoted by $F_i(k)$, the amounts of catalyst on each tray $k = 2, ..., N-1$ denoted by $E_{cat}(k)$ and 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.

The dimension of the model depends on the number of trays. The cardinality of the set of design variables is $|D| = 3N$, whereas the cardinality of the set of model variables M is $|M| = 149N + 14$.

In Figure 2 the multi modality of the design problem at hand is shown. Each of the points displayed represents one of the local optima known so far for the column design problem with fixed numbers of trays $N = 10, ..., 60$ in the space of the cost and of the revenues. The plotted line identifies the threshold between profit and loss. It can be recognized that the quality of the local optima and the composition of the cost and of the revenues of solutions which give approximately the same profit are varying significantly. These variations reflect the differences of the design and the model variables at the local optima. The computation time for one local optimization of the model with $N = 60$ trays, started from a trivial starting point in the space of all variables, is about 30 min.

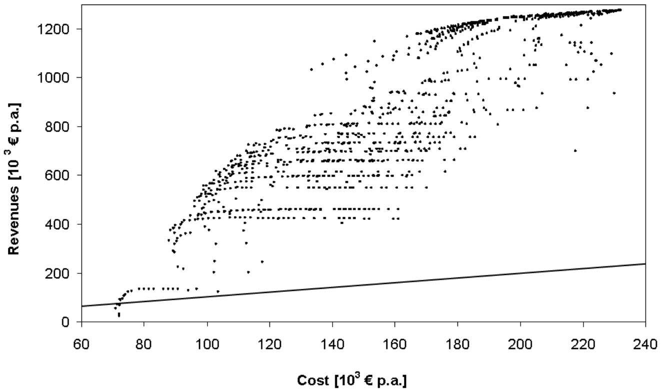

Fig. 2. Local optima of the column design problem in the space of the cost (for $N = 10, ..., 60$)

For a more detailed description of the design problem at hand see [8] and [9].

A. NLP model

The optimization problem is modeled as an NLP optimization problem. All variables (model variables and design variables) are free decision variables $X = M \cup D$ in the optimization model (in the following denoted by $MTBE_{NLP}$). This model consists of a large number of algebraic equations formulated in the modeling language GAMS [3]. These equations correspond to the set of equations (2) for the computation of the mass transfer and the reaction on each tray of the column (the so called *model equations*) and for the computation of the annualized cost.

Furthermore it comprises a set of inequalities (3) to

- fulfill the required purity of the product and the required throughputs and
- restrict operating conditions to feasible choices.

B. Simulation model

The simulation model comprises a subset of the equations (2) and the inequalities (3) of the optimization model. The design variables were removed from the set of decision variables, that is $X = M$. The equations and inequalities that restrict the feasible values of the design variables were also removed from the set of constraints (2) and (3).

One of the constraints couples the diameter Dia with the E_{cat} values and some model variables. To eliminate the problem that the equations in the model cannot be fulfilled for fixed design variables because of this constraint, the diameter and the related constraint were removed from the simulation model and are handled by the design optimization.

III. The memetic algorithm

The two main components of the memetic algorithm are an evolutionary strategy as described in [6] and the local NLP solver CONOPT [7]. In Figure 3 the structure of the memetic algorithm is shown. At the beginning of each optimization, the mathematical solver CONOPT is started from a trivial initial point, where all variables have the value 1. This first local search is necessary to find initial values for the model variables within the simulation model, because the initialization is crucial for the solution of the model equations. The values of the model variables at the first local optimum serve as initial values for the simulation in the following search procedure.

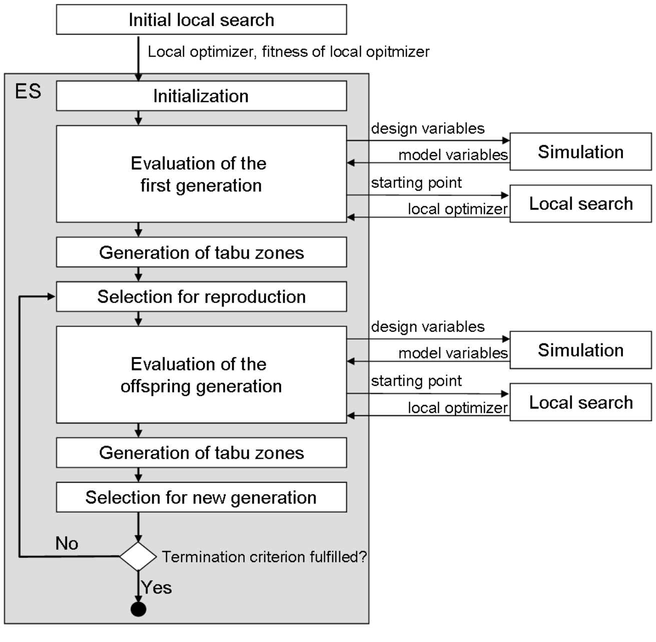

Fig. 3. Structure of the memetic algorithm (MA)

Subsequently the ES is started. The starting population of size μ is initialized randomly within the search space of the design variables with respect to the constraints as described in the next section.

To evaluate the individuals of the initial population, the simulation model is solved for each individual. If the local solver cannot fulfill the model equations, it aborts at an infeasible point. An analysis revealed, that the violation of the constraints at these points usually are so small, that these points nonetheless can be used as starting points for the subsequent local search. Then, the constraint coupling the diameter Dia, some of the model variables and the E_{cat} values which was removed from the simulation model before (see Section II-B) is checked. In case that this constraint can be satisfied, Dia is assigned to the minimal feasible value of the diameter. Otherwise it is assigned to a random value within its bounds.

The resulting points in the space of all variables are passed to the MP solver in order to perform a local search. 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 optima.

For each localized local optimum, a region is defined (a *tabu zone*) to approximate the basin of attraction of this optimum. These regions are excluded from the subsequent search procedure.

To create the λ offspring, λ times one individual is chosen randomly from the population. Each of these individuals is mutated by a modified variant of the standard mutation operator described in [10]. A self-adaptive step size introduced by [6] is applied. The initial step size is denoted by ς and is chosen with respect to the size of the feasible domain of the corresponding variable. A detailed description of the mutation operator is given below.

Then the simulation is called for the resulting offspring individuals, and the local searches are started from the resulting values of the model variables. The genes of these offspring individuals are replaced by the design variables of the corresponding local optimum. For each of these local optima a tabu zone is defined and the list of tabu zones is expanded.

If the termination criterion (here the computation time limit or a maximal number of local solver calls) is not fulfilled, the generation cycle is started again with the selection of parent individuals for the next generation of offspring. A $(\mu + \lambda)$ selection, which is advantageous in discrete search spaces is chosen as introduced in [10]. (Each individual of the population represents a local optimum and the set of local optima is discrete and finite.) The μ best individuals with respect to the objective function among all individuals that do not exceed a maximal 'life-span' of κ generations are selected to form the population of the next generation.

A. Representation

To exploit the reduction in the size of the search space, the design variables of the problem at hand are taken as the representation of the individuals. They constitute the genotype space. The genes of an individual are represented by the vector of continuous variables g:

$$g = \quad (F_1(1), ..., F_1(N), F_2(1), ..., F_2(N), E_{cat}(2), ..., E_{cat}(N-1), \alpha_{top}, \alpha_{bottom})$$

with

$$\begin{aligned} 0 \leq F_1(k) \leq 6.375,\ \forall k, \\ 0 \leq F_2(k) \leq 8.625,\ \forall k, \\ 0 \leq E_{cat}(k), \quad k = 2, ..., N-1, \\ \alpha_{top}, \alpha_{bottom} \in (0,1) \end{aligned}$$

The feasible domain of a variable x is denoted by $[lb_x, ub_x]$, where lb_x and ub_x are the lower and upper bounds of the variable. There are two constraints with respect to the design variables that have to be fulfilled in order to get a feasible solution.

$$\sum_{k=1}^{N} F_1(k) \quad = \quad F_1^{sum}, \tag{4}$$

$$\sum_{k=1}^{N} F_2(k) \quad = \quad F_2^{sum}, \tag{5}$$

with $F_1^{sum} = 6.375$ and $F_2^{sum} = 8.625$. In order to create and to maintain feasible individuals, these constraints are taken into account within the initialization procedure and the mutation procedure, as explained in the next two sections.

B. Initialization

To create an initial population of feasible individuals the constraints (4) and (5) have to be taken into account. Therefore the variables $F_1(k)$ and $F_2(k)$ are initialized in a random sequence with random values, chosen from a uniform distribution within the feasible domain of the variables and with respect to the given values of their sums F_i^{sum}. The last variables are assigned values that fulfill the given constraints.

All other variables are initialized randomly within their bounds.

C. Mutation

To maintain the feasibility of the individuals, the mutation also has to be performed respecting the constraints (4) and (5). Hence the standard mutation is modified for the feed stream variables $F_1(k)$ and $F_2(k)$ and their step sizes. The second aspect that has to be taken into account is the existence of the tabu zones.

1) Mutation of the feed streams: Both feed streams are mutated independently. One of the N indices k is chosen randomly from a uniform distribution, and the value of the corresponding stream $F_i(k)$ is changed by a normal distributed random number within its bounds $[lb_{F_i(k)}, ub_{F_i(k)}]$. In order not to leave the feasible domain of the variable, reflection is used at the bounds. Therefore the maximal step size σ_{max} is set to $\sigma_{max} = |ub - lb|/2$ for each variable. The resulting variables do not fulfill the constraints (4) or (5) any more.

In order to meet the constraints, a repair procedure is started until the difference of the sum of the amounts of all feed streams and F_i^{sum}, Δ_F, equals zero again. One of the remaining indices of the considered feed stream is chosen randomly with a uniform distribution. The value of the corresponding stream is changed, so that either $\Delta_F = 0$ or the value meets one of the bounds $lb_{F_i(k)}$ or $ub_{F_i(k)}$. If $\Delta_F = 0$, the feasibility is obtained. Otherwise the repair procedure is repeated for other untreated feed stream variables until the repair is done successfully.

The mutation procedure described above changes only one of the variables - that one chosen first - randomly. The other variables are changed to maintain feasibility. Therefore the vector of step sizes contains only one value for each feed stream.

2) Respecting the tabu zones: In order to respect the tabu zones, which are excluded from the search procedure, the offspring are not allowed to be members of the zones.

Therefore each offspring that is generated by the described mutation procedure which belongs to a forbidden zone is mutated until it leaves the zones. If this is not possible within $3 * |D|$ mutations of the object variables, the step size is mutated and the mutation cycle of the object variables is restarted. This procedure is repeated up to 100 times. If each offspring produced by the mutation is in a forbidden zone, another parent individual is chosen randomly and the procedure is started again up to 10 times. If no offspring outside the tabu zones can be generated, the overall optimization procedure is stopped.

D. The tabu zones

In order to approximate the basins of attraction, tabu zones of different forms and sizes and with different locations in relation to the corresponding local optima were tested. Tabu zones with two different forms (rectangular and spherical) were examined. The locations of the starting points and of the corresponding local optima define the locations of the tabu zones. All examined zones are symmetric (for each coordinate) around their centers. If the local optimum is taken as the center, the tabu zones are called *symmetric*. Their size is determined by the distance between the starting point and the corresponding local optimum. If the midpoint between the starting point and the optimum is taken as the center, the zones are called *asymmetric*. The four different types of zones that were investigated are shown in Figure 4 schematically in a two-dimensional space. To vary the size of the tabu zones, a scaling factor β is defined. A scaling value of $\beta = 1$ corresponds to the distance between the starting point and the local optimum. Before the computation of the radius of the spherical tabu zones, the values of the variables were scaled by dividing them by a characteristic value. This value is chosen to be either the value in the middle of the feasible domain (for the bounded variables) or the result of the first local optimization (for the unbounded variables).

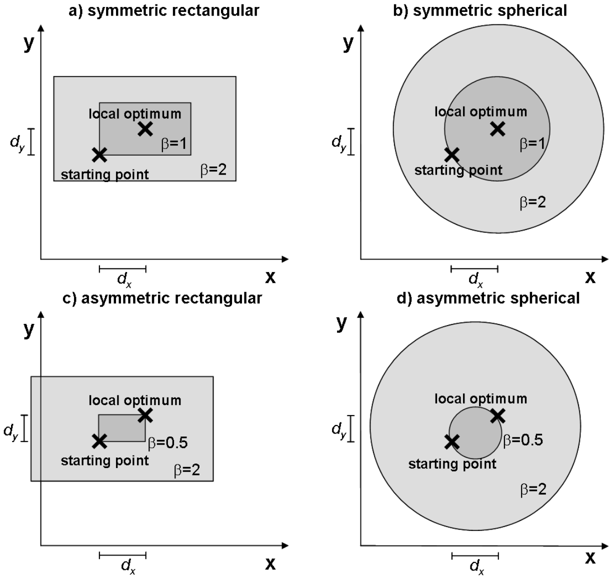

Fig. 4. Schematic structure of the tabu zones

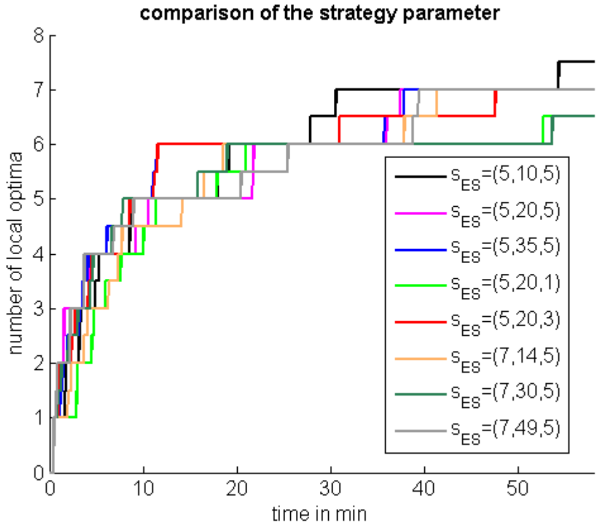

Fig. 5. Median progress curves of the MA using different strategy parameters, $s_{ES} = (\mu, \lambda, \kappa)$

TABLE I

RESULTS OF THE STRATEGY PARAMETER TESTS - NUMBERS OF THE LOCAL OPTIMA FOUND WITHIN ONE COMPUTATION HOUR

s_{ES}	best	1st quart.	median	3rd quart.	worst
(5,10,5)	**9**	**8**	**7.5**	**7**	**6**
(5,20,5)	8	7.75	7	7	6
(5,35,5)	8	7	7	7	6
(5,20,1)	8	7	7	7	6
(5,20,3)	8	7	7	6.25	5
(7,14,5)	8	7.75	7	7	6
(7,30,5)	7	7	6.5	6	6
(7,49,5)	8	8	7	7	6

IV. EVALUATION

To test the memetic algorithm, a parameter tuning of the strategy parameters $s_{ES} = (\mu, \lambda, \kappa)$ for the ES was done (see Section IV-A). Furthermore the memetic algorithm was tested using different tabu zones and different scaling factors β (see Section IV-B).

In Section IV-C the results of test runs of the MA optimizing more complex models with a time limit of 24 hours are discussed.

Finally, the memetic algorithm with the best settings was compared to a reference algorithm (OQNLP/CONOPT) that was used to solve the MINLP design-optimization problem of the MTBE column before [8]. The reference algorithm is described in Section IV-D.1 and the results are given in Section IV-D.2.

In all of the following test runs, the initial step size is chosen to be $\varsigma = |ub - lb|/2$ for each variable.

A. Strategy parameter tests

In order to find good strategy parameters $s_{ES} = (\mu, \lambda, \kappa)$ for the ES, the MA was tested 10 times with $N = 10$ trays using the symmetric rectangular tabuzone and a scaling factor $\beta = 1.5$ for each of the following parameter settings:

$$\{(5, 10, 5), (5, 20, 5), (5, 35, 5), (5, 20, 1),\\ (5, 20, 3), (7, 14, 5), (7, 30, 5), (7, 49, 5)\}.$$

In Figure 5 the median progress curves of the MA using different strategy parameters are shown. Table I comprises the numbers of local optima found within one hour of computation time (all four quartiles are listed).

It can be seen that the best overall progress and the largest median number of local optima found was reached by using the strategy parameters $s_{ES} = (5, 10, 5)$. It is the only parameter vector tested, where all known local optima were found within the given time in the best run. The median value of the time needed to find 7 local optima is 30.5 CPU min, that is 18% less than the second best result of using the parameters $s_{ES} = (5, 20, 5)$.

In the following, the strategy parameter vector is fixed to $s_{ES} = (5, 10, 5)$.

B. Evaluation of the tabu zones

The application of the four types of tabu zones as described in Section III-D were tested with different scaling factors β. The memetic algorithm was tested 10 times for each tabu zone and each scaling factor, as well as the memetic algorithm without the tabu zones.

The aim of the optimization of the problem at hand is to find the global optimum. The first (and the most dominant) criterion that is taken into account to choose the best scaling factor is therefore the median value of the total number of local optima found. Because of the complexity of the problem at hand, the second criterion is the number of local solver calls that is needed to find these median number of local optima. And finally the best and the worst numbers of local optima found should be as large as possible.

Figure 6 shows how the scaling factor β influences the best, median and worst values of the numbers of local optima found within one hour of computation. 9 local optima are known so far for the 10-trays model. $\beta = 0$ corresponds to the results of the MA without any tabuzone.

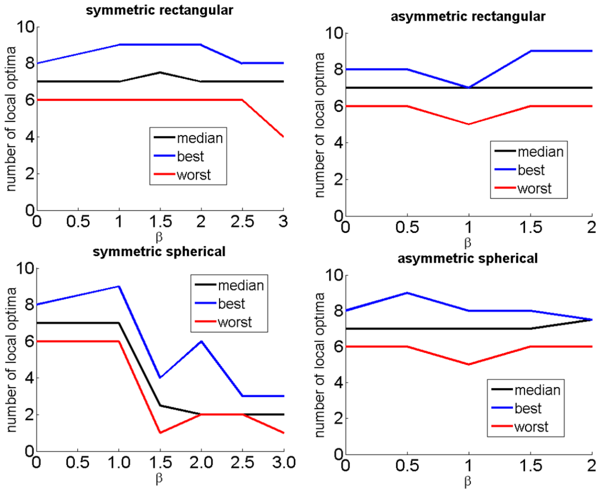

Fig. 6. Influence of the scaling factor β on the numbers of local optima found, $s_{ES} = (5, 10, 5)$

The median values of the number of local solver calls needed to find 7 optima is listed in Table II. The number in brackets is the fraction of the test runs with less than 7 local optima found.

TABLE II

RESULTS OF THE PARAMETER TESTS FOR THE DETERMINATION OF THE BEST SCALING FACTORS β FOR THE TABU ZONES - SOLVER CALLS AND FRACTION OF RUNS WHERE LESS THAN 7 OPTIMA WERE FOUND, $s_{ES} = (5, 10, 5)$

β	without tabu zones	symmetric rectangular	symmetric spherical	asymmetric rectangular	asymmetric spherical
0.5	**266** **(30%)**	-	-	273 (20%)	252 (30%)
1.0		229 (30%)	**209** **(40%)**	308 (40%)	224 (40%)
1.5		**193** **(15%)**	- (100%)	**183** **(10%)**	279 (10%)
2.0		336 (35%)	- (100%)	260 (30%)	**166** **(30%)**
2.5		224 (35%)	- (100%)	-	-
3.0		285 (40%)	- (100%)	-	-

For the symmetric rectangular tabu zones, the best scaling factor is $\beta = 1.5$. By using this scaling factor the best median number of local optima found was reached and the lowest number of local solver calls was needed to find 7 local optima. The fraction of runs, where less than 7 local optima were found, was reduced to 15%.

The best scaling factor for the asymmetric rectangular tabu zones is also $\beta = 1.5$. Although the best, median and worst numbers of local optima found is equal by using $\beta = 2.0$, the median number of solver calls needed to find 7 local optima and the fraction of runs with less than 7 local optima are significantly smaller by the use of $\beta = 1.5$.

The application of the symmetric spherical tabu zones does only improve the optimization procedure by using a scaling factor of $\beta = 1.0$. Scaling factors of $\beta > 1$ lead to large tabu zones, that cover the whole feasible domain of some of the variables, so that the optimization procedure is aborted after one to four generations. That is attributed to the scaling procedure, because the E_{cat} values are unbounded. The characteristic value which is used to scale the variables is a realistic value at a local optimum, but it is too small for some of the starting values proposed by the ES.

For the asymmetric spherical tabu zone, $\beta = 2.0$ is the factor that leads to the smallest medium number of solver calls needed to find 7 local optima and to the largest median number of local optima found. Although the fraction of runs, where less than 7 optima were found is 30% in comparison to 10% by using $\beta = 1.5$, the significantly smaller number of solver calls is the most dominant factor, because of its influence on the progress curves of the MA over the time.

In comparison to the results of the MA without tabu zones, the median value of the number of solver calls that is needed to find 7 local optima was improved by the introduction of the asymmetric rectangular tabu zones by 32% and the fraction of the optimizations with less than 7 located optima is reduced by 2/3. In case of the asymmetric spherical tabu zone the number of solver calls was improved by even 38%, but the fraction of the optimizations with less than 7 located optima was not improved.

In order to compare the different types of tabu zones, the median progress curves of the MA for each zone using the best scaling factor is plotted in Figure 7. It shows the median number of local optima found at each point of time of the optimization.

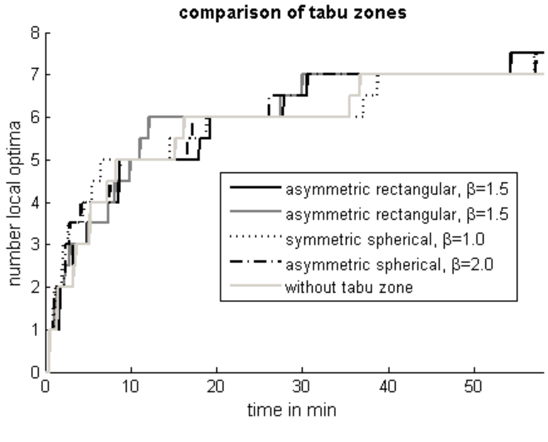

Fig. 7. Median progress curves with the tabu zones using the best tested scaling factors β, $s_{ES} = (5, 10, 5)$

The best median number of local optima found within the given time limit is 7.5 and was reached by the application of the symmetric rectangular tabu zones and by the application of both spherical tabu zones. The optimization using the spherical tabu zones show the fastest progress at the beginning, but if the overall progress is taken into account, the symmetric rectangular tabu zones approximate the basins of attraction best.

A closer look at the progress curves reveals, that the introduction of the symmetric rectangular tabu zone using a scaling factor $\beta = 1.5$ reduced the time to find 7 local optima by 29% and increased the median number of local optima found within one hour of computation by 5.5%. The fraction of runs with less than 7 located optima was reduced by 2/3.

C. Analysis of the MA

In order to study the behavior of the MA for difficult problems, complex models with $N = 40$ and 60 trays were optimized using a time limit of 24 hours. The symmetric rectangular tabu zones with a scaling factor of $\beta = 1.5$ and the strategy parameter vector $s_{ES} = (5, 10, 5)$ were used. Figure 8 shows the progress curves of the test runs. At the beginning of the optimization of the 40-trays model, a fast progress can be observed. With proceeding time the progress decreases and after 392 solver calls no new local optimum can be found by the MA. Until this point of stagnation every fifth to sixth local solver call on the average leads to a new local optimum. The progress of the optimization of the 60-trays model stagnates after 249 local solver calls. Up to this call every fourth to fifth local solver call leads to a new local optimum.

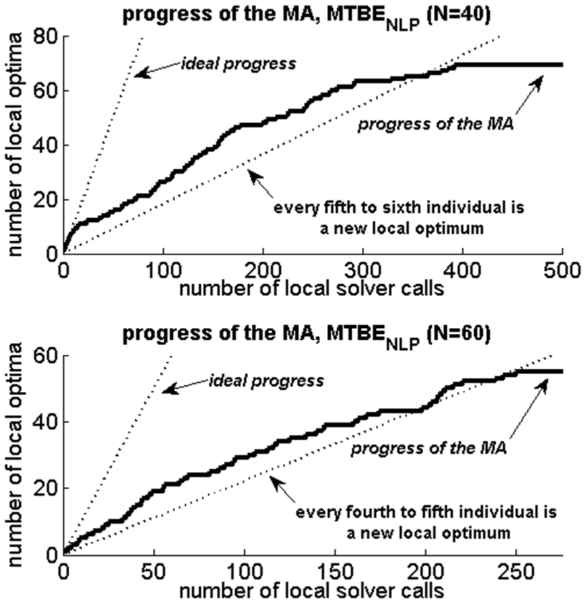

Fig. 8. Progress curves of the MA for the 40-trays and 60-trays model, $s_{ES} = (5, 10, 5)$

In order to find out why the progress stagnates, in Figures 9 and 10 the compositions of the parent populations of the generations are displayed. Here all local optima known so far were successively indexed ordered by the values of the objective function.

Figures 9 and 10 reveal that there is a loss of diversity in the population of the parents during the optimization process. The parent populations of all generations consist of only 11 different local optima (80 optima are known so far) in case of the optimization of the 40-trays model and of only 5 optima (out of 55 optima known so far) in case of the 60 trays-model. For the 40-trays model each population of the generations 50 to 120 consist of $\mu = 5$ individuals that represent the same local optimum. The selection of only the best individuals for the next generation prevents that the parent populations contain inferior individuals and it does not take into account that several individuals in one population may represent the same local optimum. Other selection procedures will be tested in the future that accept inferior individuals and that take the diversity into account.

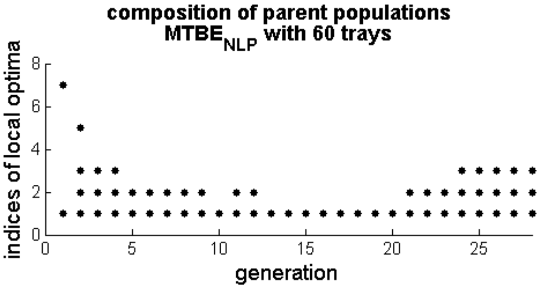

Fig. 9. Compositions of the parent populations of the MA for the 60-trays model

In Figure 11 and 12 the compositions of the offspring populations of the MA can be seen. Although there is a severe loss of diversity in the parent populations, Figure 11 and Figure 12 show that the MA covers a big part of the search space during the whole optimization procedure.

D. Comparison with the reference algorithm

The results of the MA were compared to the performance of a reference algorithm that was used before to optimize the overall MINLP problem of the case study at hand [8]. Both algorithms were tested on a PC with 1.8 GHz and 1GB of memory. 20 runs of each algorithm were executed for the $MTBE_{NLP}$ model with $N = 10$, 15, 40 and 60 trays.

1) The reference algorithm: OQNLP [11] is a scatter search based multi-start heuristic, that generates different starting points for a local MP 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 [7]. As in the MA, the solver optimizes in the space of all variables.

To allow a fair comparison of both algorithms, a parameter tuning was done for the reference algorithm too (see [12]).

2) Comparison of results: Two criteria are used to measure the performance of the algorithms. The first criterion is the quality of the starting points for the local search, proposed by the MA and the multi-start heuristic OQNLP. This quality is measured by the cpu seconds the local solver needs to find a local optimum starting at the given point. The second criterion is the performance of the global search, that is the quality of the best local optimum found and the number of localized optima.

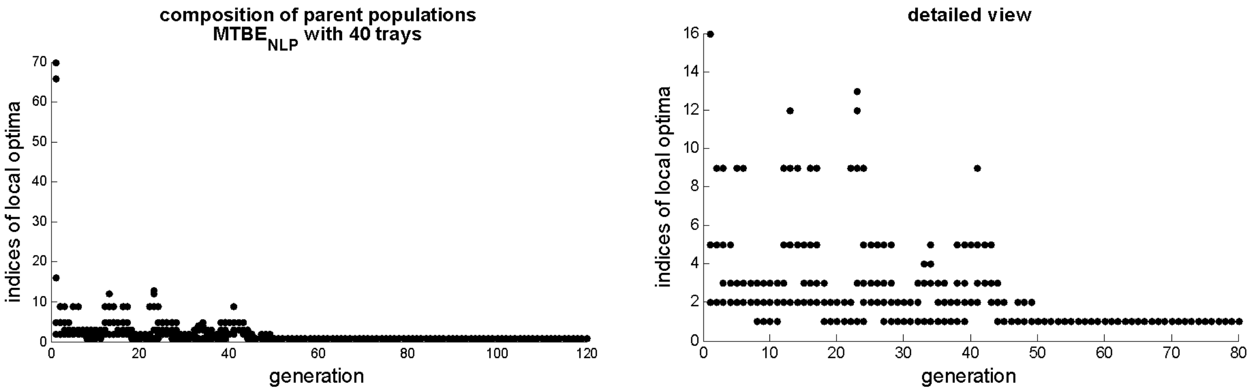

Fig. 10. Compositions of the parent populations of the MA for the 40-trays model

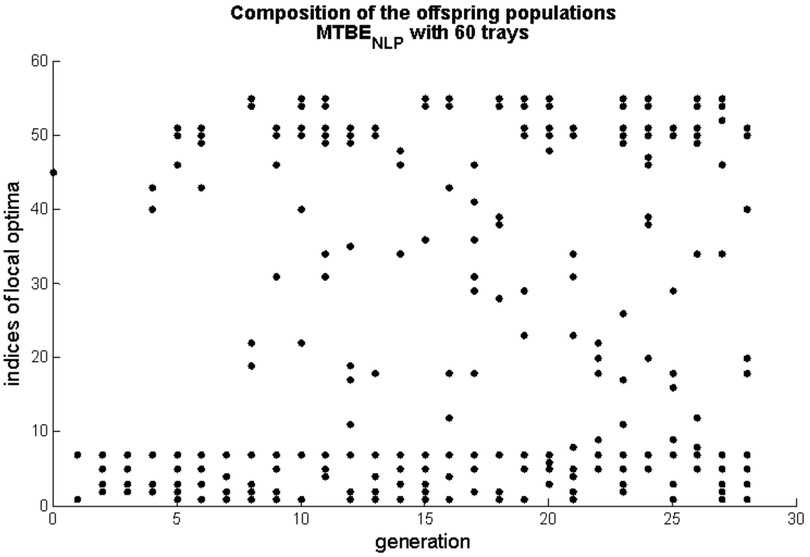

Fig. 11. Compositions of the offspring populations of the MA for the 40-trays model

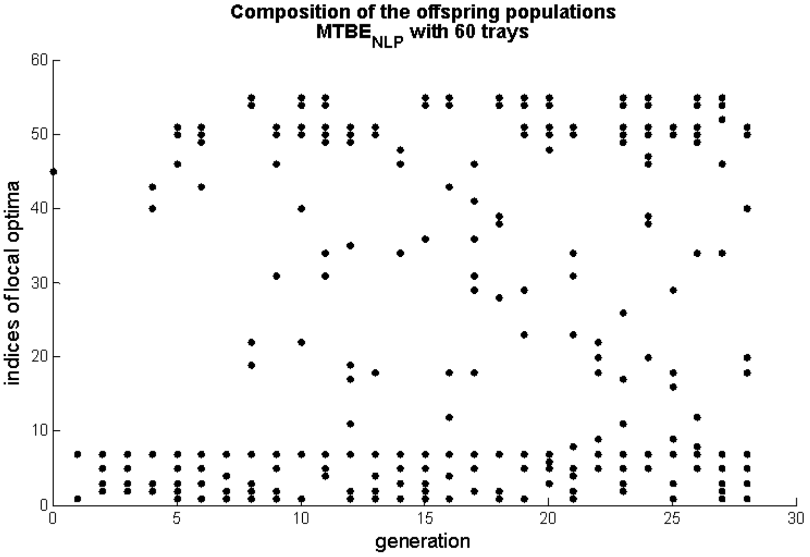

Fig. 12. Compositions of the offspring populations of the MA for the 60-trays model

TABLE III

Best, median and worst values of computation times of the local solver CONOPT and ratios of infeasible aborts ($MTBE_{NLP}$ with $N =$ 10, 15, 40 and 60 trays)

N		OQNLP/CONOPT		MA	
		time [s]	infeasible aborts [%]	time [s]	infeasible aborts [%]
10	min	23	7.3	4	15.8
	median	34	14.2	9	18.5
	max	50	21.2	17	22.0
15	min	58	2.3	13	11.2
	median	85	16.5	20	30.6
	max	128	27.3	34	61.1
40	min	609	9.1	58	0.0
	median	737	33.3	116	0.0
	max	915	69.2	223	3.6
60	min	1586	11.1	180	0.0
	median	1896	31.7	291	2.2
	max	2211	61.5	492	8.9

In Table III, the best, the median and the worst values of the computation times of CONOPT and the ratio of the local solver calls that were aborted without finding a feasible solution to the total number of local solver calls are listed. The times for the MA comprise the times for both CONOPT calls, the simulation and the subsequent local optimization, whereas the times for OQNLP/CONOPT are times of the local search. The results show that the reduction of the search space and the initialization of the simulation model could improve the quality of the starting points considerably. The computation time needed for the local search with starting points proposed by OQNLP was reduced by more than 75% by the application of the MA. In the optimization runs of the more complex models ($N = 40, 60$), the ratio of the infeasible aborts of the local search to the total number of all local solver calls was reduced to a value of less than 3% (median), even in the worst case it is less than 10%. This ratio is around 30% for the application of OQNLP/CONOPT and shows large fluctuations. The median ratio of 30.6% of the MA in comparison to 16.5% of OQNLP/CONOPT for the optimization of the model with $N = 15$ is a result of the significantly greater number of solver calls of the MA within the given time of 60 cpu min for the optimization runs.

Figure 13 shows the best, the median and the worst progress curves of the algorithms with regard to the best objective found during the optimization for columns with 15 and 60 trays. The termination criterion for the optimization was a time limit of 1 hour for the 15-trays model and a limit of 4 hours for the 60-trays model.
For the model with $N = 15$ trays, the time to find the best local optimum known so far was 187 sec (median), 99 sec (best) and 450 sec (worst) for the MA and 419 sec (median), 164 sec (best) and 1524 sec (worst) for OQNLP/CONOPT. For the model with $N = 60$ trays, the MA found the best local optimum known so far after 94 min (median), 36 min (best). In only 1 of the 20 runs this optimum was not found within 4 hours. OQNLP/CONOPT needed 125 min in the best run and could not find the best optimum in 14 of 20 runs within the given time limit.

In Figure 14, the best, the median and the worst numbers of local optima found so far are plotted for each point in time of the optimization runs. The results show that the maximal number of localized optima during the optimization process could be increased significantly by the MA in all cases. A closer look at the results of the optimization of the 60-trays model reveals that even the worst run of the MA was much better than the best optimization run of OQNLP/CONOPT.

V. Conclusion and Outlook

In this paper, the development and the analysis of a memetic algorithm for the optimal design of chemical processes was described. The application of an evolutionary strategy in combination with a mathematical solver based on the generalized reduced gradient method (CONOPT) enabled the reduction of the search space to the set of design variables. A good initialization procedure for the simulation that is needed to compute the cost for given values of design variables leads to a significant reduction of the computation time for the following local search in comparison to the reference algorithm OQNLP/CONOPT, which works in the space of all variables. The introduction of tabu zones, which approximate the basins of attraction of the local optima found so far and their exclusion from the subsequent search procedure improved the performance of the MA by 29%.

The detailed analyses of the behavior of the MA showed that there is a loss of diversity in the ES because of the selection of only the best individuals to survive. Strategies to maintain (or reinstall) the diversity of the population during the search procedure will be examined in future work.

The development of a step size adaptation rule with respect to the fact that all individuals represent local optima of the problem will also be investigated. The self adaptation method used here guides the search to promising areas in the search space and it enables to adapt the step size depending on the topology of the neighborhood. The new method should respect the approximated basins of attraction and it should guide the search to the unexplored regions of the search space.

In future work, the model used here will be extended by a restriction of the number of feed streams. In practice, only a small number of feed streams is used. This restriction is difficult to handle in an MINLP approach.

Acknowledgment

This research was supported by the DFG Collaborative Research Centre ’Design and Management of Complex Technical Processes and Systems Using Computational Intelligence Methods’ (Sonderforschungsbereich 531) at Technische Universität Dortmund. This support is very gratefully acknowledged.

References

[1] K. Sundmacher and A. Kienle, *Reactive Distillation – Status and Future Trends*, Whiley-VCH: Weinheim, Germany, 2003.

[2] A. W. Westerberg, “A retrospective on design and process synthesis”, *Computers & Chemical Engineering* 28, pp. 447–458, 2004.

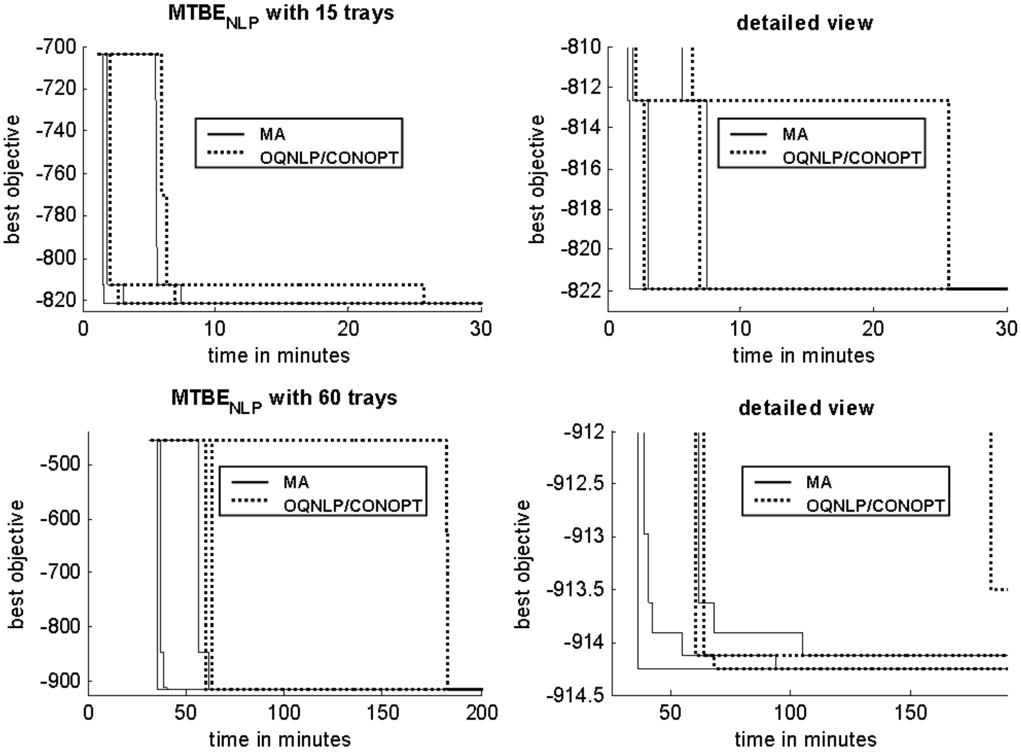

Fig. 13. Best, median and worst progress curves of the MA and OQNLP/CONOPT

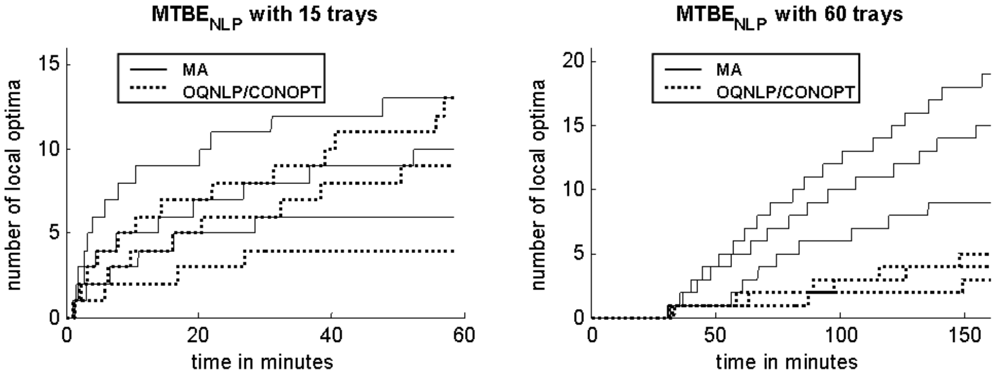

Fig. 14. Best, median and worst number of local optima found by the MA and OQNLP/CONOPT

[3] A. Brook, D. Kendrick, A. Meeraus and R. Raman, *GAMS - A User's Guide*, Washington: GAMS Development Corp., 1998.

[4] A. R. Ciric and D. Gu, "Synthesis of nonequilibrium reactive distillation processes by MINLP Optimization", *AIChE J.* 40, pp. 1479–1487, 1994.

[5] J. Stichlmair and T. Frey, "Mixed-integer nonlinear programming optimization of reactive distillation processes", *Ind. Eng. Chem. Res.* 40, pp. 5978–5982, 2001.

[6] H.-P. Schwefel, *Evolution and Optimum Seeking*, New York: Wiley Interscience, Sixth-Generation Computer Technology, 1995.

[7] A. S. Drud, "CONOPT: A large-scale GRG code", *ORSA Journal on Computing* 6, pp. 207–216, 1992.

[8] S. Barkmann, G. Sand and S. Engell, "Modellierungsansätze für die Design-Optimierung von reaktiven Rektifkationskolonnen", *Chemie Ingenieur Technik* 80, pp. 107–117, 2008.

[9] G. Sand, S. Barkmann, S. Engell and G. Schembecker, Structuring of reactive distillation columns for non-ideal mixtures using MINLP-techniques, *European Symposium on Computer-Aided Chemical Engineering* 14 (18), pp. 493–498, 2004.

[10] H. G. Beyer and H. P. Schwefel, "Evolution Strategies - a comprehensive introduction", *Natural Computing: An International Journal* 1(1), pp. 3–52, 2002.

[11] F. Glover, M. Laguna and R. Marti, "Scatter search" in *Advances in Evolutionary Computation: Theory and Applications*, Edited by A. Ghosh and S. Tsutsui, pp. 518–537, Springer, 2003.

[12] G. Sand, S. Barkmann, P. Scherpian, M. Tylko, S. Engell and G. Schembecker, MINLP-Optimization in the Integration of Reaction and Separation Processes, in Sustainable (Bio)Chemical Process Technology, pp. 317–324, 2005.