 https://doi.org/10.3311/PPch.12021

Periodica Polytechnica Chemical Engineering, 63(1), pp. 210–225, 2019

Hierarchical Representation Based Constrained Multi-objective Evolutionary Optimisation of Molecular Structures

Gyula Dörgő[1*], János Abonyi[1]

[1] MTA-PE Lendület Complex Systems Monitoring Research Group, Department of Process Engineering, University of Pannonia, H-8200 Veszprém, P.O.B. 158, Hungary
[*] Corresponding author, email: gydorgo@fmt.uni-pannon.hu

Received: 30 January 2018, Accepted: 06 April 2018, Published online: 14 June 2018

Abstract

We propose an efficient algorithm to generate Pareto optimal set of reliable molecular structures represented by group contribution methods. To effectively handle structural constraints we introduce goal oriented genetic operators to the multi-objective Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The constraints are defined based on the hierarchical categorisation of the molecular fragments. The efficiency of the approach is tested on several benchmark problems. The proposed approach is highly efficient to solve the molecular design problems, as proven by the presented benchmark and refrigerant design problems.

Keywords

structural optimisation, genetic operators, hierarchical constraints, similarity of Pareto Fronts

1 Introduction

Computational intelligence has opened a new way to the design of chemical technologies from the smaller scale of molecules to the bigger scale of process design, which is consistent with the expectations of Industry 4.0 and with the trends of modern chemical engineering [1-3]. Nowadays, numerous computational methods are available for the solution of the computer aided molecular design (CAMD) task. Neural networks are mainly used for the revealing of correlations between structural and macroscopic properties [4-6]. The design task is usually handled as an optimisation problem. Among the wide range of applicable algorithms particle swarm optimization (PSO) is used for drug design [7, 8]. Linear programming is also applicable [9, 10], but usually the design task is formalised as mixed integer nonlinear integer programming (MINLP) problem. Among the wide range of computational intelligence related solvers simulated annealing and outer approximation programming algorithm have been applied [11]. As it is illustrated by Mitra [12] and Weber [13] chromosomes of genetic algorithms (GAs) are convenient ways to represent molecular structures, thus most of the successful approaches use different types of genetic algorithms e.g. in [14-17].

GAs are popular global optimization algorithms developed by Holland [18] based on the mimicking of natural selection and the competitive model of survival of the fittest. The modification of genetic operators to suit the specific problem seems to be a promising way for the solution of large, complex problems, there is a trend towards exploring and developing such algorithms [19, 20].

Michalewicz has showed that repair functions and special genetic operators are highly effective for the solution of constrained genetic optimization problems [21]. Renner and Ekárt also reported the effectivity of such functions in computer aided design [22]. Therefore, numerous publications can be found in literature with special and goal-orientedly modified genetic operators. Deep and Thakur introduced new mutation and crossover operators for real coded genetic algorithms [23, 24]. Colanzi and Vergilio could improve the previous empirical studies for the optimization of product line architectures with the implementation of a modified crossover operator [25]. Strug et al. represented the genotypes of design problems with hierarchical hypergraphs and introduced modified mutation and cross-over operators to adapt to such representation [26]. Salimi et al. solved scheduling and load balancing optimization problems with the application of fuzzy adaptive operators [27].

Venkatasubramanian et al. introduced a modified genetic algorithm with string representation of the molecular structures and new, task-oriented genetic operators that

Cite this article as: Dörgő, Gy., Abonyi, J. "Hierarchical Representation Based Constrained Multi-objective Evolutionary Optimisation of Molecular Structures", Periodica Polytechnica Chemical Engineering, 63(1), pp. 210–225, 2019. https://doi.org/10.3311/PPch.12021

facilitate the chemistry of molecular interactions and rearrangements [28]. Dyk and Nieuwoudt applied new genetic operators during the design of solvents for distillation processes [29]. Glen and Payne described a genetic algorithm with 12 task-oriented mutation operators for the modification of the molecular structure used in the automated generation of molecules within constraints [30]. Brown et al. proposed novel genetic operators for the modification of graph-based representation of molecular structures [31].

We believe that the biggest challenge of evolutionary molecular design is that several conflicting objectives must be handled simultaneously. The application of multi-objective genetic algorithms is a promising approach to generate the Pareto set of solutions which represent different design aspects. We demonstrated that the well-established multi-objective Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is ideal tool to handle this problem [32]. There is a need to further improve this approach by increasing the efficiency of the search in the huge chemical search space represented by group contribution methods [33].

The main problem with the application of multi-objective evolutionary algorithms for molecular design is the difficulty of handling of a large number of structural constraints. Our key idea is that the efficiency of the algorithm can be significantly increased by the introduction of problem-relevant genetic operators, which always generate feasible solutions. Since in molecular design the building blocks of the optimised structure are not identical, for a parsimonious and systematic design of these constraints it is beneficial to characterise how the molecular fragments can be connected to each other and define the constraints and the related genetic operators based on the hierarchical classification of the identified connection rules.

Our key contribution is therefore that instead of testing the branching and the octet rule related feasibility constraints we introduce special genetic operators that ensure the generation of reliable and connectable molecular fragments. It is important to highlight, that in the present paper we only describe the generation of connectable fragments and not any type of spatial structure of the molecule. The huge variety of molecular segments could result in hundreds of constraints. To handle this problem, we generated the goal oriented operators based on the hierarchical categorisation of the molecular fragments ensuring the effective incorporation of chemical information into the optimisation algorithm.

From now on the formalisation of the design problem is followed by a theoretical overview of the nature of genetic algorithms paying special attention to NSGA-II. After the description of the different algorithms proposed for the solution of the design task, the efficiencies of these approaches are examined through two design problems, and the results are discussed extensively to determine the improvement in the applicability of these algorithms and to compare the effectiveness with other approaches from the literature.

The proposed method was implemented in MATLAB. The results are reproducible since all the functions and numerical experiments are downloadable from the website of the authors: www.abonyilab.com.

2 Problem formulation

2.1 Hierarchical representation of molecular structure

Our aim is to design molecules that simultaneously meet several requirements, so we deal with models that can estimate m properties $\boldsymbol{P}(\boldsymbol{x}) = \left[P_1(\boldsymbol{x}), P_2(\boldsymbol{x}), \ldots, P_m(\boldsymbol{x})\right]^T$. Each property is estimated based on a group contribution method realised by a function $P_k = f_k(x_1, x_2, \ldots, x_n)$ $(k = 1, 2, \ldots, m)$ (here we would like to mention that there are important thermodynamic properties that are state functions and therefore could not be predicted by group contribution methods). Therefore, the molecule is represented by an n dimensional vector of positive integers $\boldsymbol{x}$, as $\boldsymbol{x} = \left[x_1, x_2, x_3, \ldots, x_n\right]$ (where $x_1, x_2, \ldots, x_n$ are the number of group type #1, #2, …, #n, respectively). For simpler handling, the fragments from the Joback method are classified according to Fig. 1. Vector **x** therefore yields the number of the assigned molecular fragments, e.g. x_i = |–O–| and $\boldsymbol{x}_2$ = |–X–|, where –X– is the set of fragments with the given structure (–X–={–CH_2–,–O–}).

2.2 Formulation of the multi-objective optimisation problem

During a material design task, multiple objectives must be taken into consideration simultaneously. The solution of such problems can be computed by combining them into a single criterion to be optimised, called a *utility function* [34]. Considering the characteristics of design problems, some of the specified features need to be maximised,

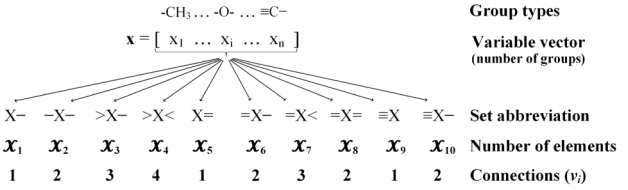

Fig. 1 The classification of the Joback molecular fragments based on the type and number of bonds

minimised or close to the given targets, as described by U_q in Eq. (1), the utility function of target q ($\widehat{P}_q$ stands for the target value of target q).

$$U_q = \begin{cases} \max P_q(\boldsymbol{x}) \\ \min P_q(\boldsymbol{x}) \\ \min\left(P_q(\boldsymbol{x}) - \widehat{P}_q\right)^2 \end{cases} \quad (1)$$

Consequently, the target properties are incorporated into a set of utility functions, $\boldsymbol{U}$, an nq dimensional vector as $\boldsymbol{U} = \left[U_1(\boldsymbol{x}), U_2(\boldsymbol{x}), \ldots, U_{nq}(\boldsymbol{x})\right]^T$.

The general formulation of an optimal CAMD design problem supposing a multi-objective optimisation problem with nq functions is presented in Eq. (2):

$$\max_{x \in \left(\mathbb{Z}^{+n} \cap \Omega\right)} \left\{U(\boldsymbol{x}) = \left[U_1(\boldsymbol{x}), U_2(\boldsymbol{x}), \ldots, U_{nq}(\boldsymbol{x})\right]\right\}, \quad (2)$$

subject to

$$g_s(\boldsymbol{x}) = 0$$
$$D_s(\boldsymbol{x}) \mid 2$$
$$E_s(\boldsymbol{x}) > 0$$
$$P_c^l \le P_c(\boldsymbol{x}) \le P_c^u,$$

where Ω is the space of feasible solutions, $g_s(\boldsymbol{x})$ is a set of structural equality constraints, $D_s(\boldsymbol{x})$ is a set of divisibility constraints, and $E_s(\boldsymbol{x})$ is a set of restrictions for existence. The property constraints are represented by P_c^l and P_c^u.

Due to the often conflicting objective functions, the solution of the optimisation problem is not a single solution, but a group of points called the Pareto-optimal set, which is defined as follows:

Definition 1. $\boldsymbol{x}^* \in \Omega$ is called a Pareto-optimal (efficient, non-inferior or non-dominated) solution of a multi-objective problem described in Eq. (2), if another feasible solution $\boldsymbol{x}$ does not exist such that $U_q(\boldsymbol{x}) \le U_q(\boldsymbol{x}^*)$, $\forall q \in \{1, 2, \ldots, nq\}$ and $U_p(\boldsymbol{x}) < U_p(\boldsymbol{x}^*)$ for at least one $p \in \{1, 2, \ldots, nq\}$ [35].

Therefore, the Pareto-optimal solutions are those for which improvement in one objective can only take place with the worsening of at least one other objective function. Pareto-ranking is the process of determining the rank of each solution through identifying the number of other solutions that dominate it [36].

2.3 Constraints

The solution of practical problems is often restricted by some constraints imposed on the decision variables. Explicit constraints can usually be classified into two major groups [37]:

Domain constraints defined for the expression of the definition of the objective function. In the present formalisation structural feasibility constraints are defined to ensure the solutions are restricted to the space of feasible solutions.

Preference constraints which impose further constraints based on further knowledge of the designer. In the case of material design, constraints of design aspects (property constraints and design limitations) are defined, which must be established for each specific design problem.

2.4 Structural feasibility constraints

1. The fragments of a feasible molecular structure can be joined to a single connected component, while unconnected bonds are prohibited. This means that neither connections of (-Cl, –Cl, –Cl) nor (>CH-, -Br) are feasible molecules, due to the unconnected molecular fragments and unconnected bonds, respectively. The formulation of a single molecular structure is ensured by the octet rule, as described in Eq. (3) [38]:

$$\sum_{i=1}^{n} (2 - v_i) \cdot x_i - 2 = 0, \quad (3)$$

where v_i stands for the connections of group i, that is the number of other fragments it can be connected with (for example –X, =X and ≡X fragments have 1, while –X– and =X= have two connections). The connections of each type of fragments are presented in Fig. 1.

Further constraints are applied only in the presence of multiple bond types simultaneously, e.g. when the molecular structure contains single and double and/or triple bonds as well.

2. The feasibility of the molecule further requires that each connection should have its appropriate pair, e.g. each =X should have a double bond to connect with within the molecule. This means that the sum of single, double or triple bonds of fragments in a molecule must be even, respectively. For the satisfaction of feasibility, the connectivity of different types of molecular subunits in terms of bond type must be maintained with "bridge" fragments, e.g. =X-, =X<, ≡X-. For example, if a molecule contains double and single bonds simultaneously, the connection between them must be ensured with a "bridge-like" group, marked with the bold character in X=**X**-X-X. This rule for double bonds is represented in Eq. (4).

$$(\mathcal{X}_6 + \mathcal{X}_7 - \mathcal{X}_5) \mid 2 \quad (4)$$

When the molecule contains groups with double bonds, the group =X= does not influence the

connection of fragments in the molecular structure in terms of feasibility, it can be added to the structure or removed from it without restrictions. This existence constraint is described in Eq. (5).

$$\mathcal{X}_8 > 0 \leftrightarrow \left(\mathcal{X}_5 + \mathcal{X}_6 + \mathcal{X}_7\right) > 0 \tag{5}$$

3. In the case of triple bonds, the even number of triple bond connections and the appropriate number of "bridge" groups need to maintain the link with the other bond types as described in Eq. (6).

$$\left(\mathcal{X}_{10} - \mathcal{X}_9\right) | 2 \tag{6}$$

One should notice that the number of double and triple bonds is constrained in the connection of molecular fragments, but none of the constraints restricts the number of single bonds. This is because if the octet rule and the criteria for double and triple bonds are fulfilled, then the number of single connections of the groups in a molecule must be even and the links to the subunits with double or triple bonds solved.

2.5 Constraints of design aspects

During the formalisation of preference constraints, the designer must form restrictions for the solution according to the knowledge at a higher level. The designer can define the type of available groups, their available numbers, and the constraints for physicochemical properties as well.

1. The available group types can indicate many macroscopic properties, e.g. unsaturated fragments can suggest the ability to polymerise or certain fragments can imply environmentally harmful compounds.
2. The property constraints are obtained from the character of application for the designed material, e.g. minimal boiling or freezing point. The lower $\left(P_c^l\right)$ and upper limits $\left(P_c^u\right)$ for property c are given in Eq. (7), where $\left(c = 1, 2, \ldots, nc\right)$

$$P_c^l \le P_c\left(\boldsymbol{x}\right) \le P_c^u. \tag{7}$$

As part of the solution of the above-defined CAMD task, the "generate-and-test" method can seem to be inefficient as a large number of candidate molecules are created which finally turn out to be infeasible from the view of connectivity. In the present work feasibility constraints are implemented in the algorithm to filter out the resultant molecular structures in terms of feasibility. As in this approach the candidates are still filtered out after property evaluation, the efficiency of the search can still seem to be inefficient. To solve this contradiction problem, specifically modified genetic operators are introduced which improve the individuals of the populations to ensure the estimation of properties is based on only reliable structures, and the property evaluation is carried out on solely feasible molecules.

In the present work, the Joback method used to predict specific properties, while genetic algorithms are applied for the design of candidate molecular structures. Based on the characteristics of CAMD problems the evolutionary algorithm is commonly accepted as an efficient method of finding solutions, thus to solve the problem defined in Section 2 the use of these stochastic minimum search-based algorithms was a promising approach. To achieve the Pareto-optimal solutions two major methods are common:

- single objective-based solutions with multiple applications of the approach conducted to find a set of Pareto-optimal solutions
- an evolutionary algorithm with multiple objectives, in which multiple Pareto-optimal solutions are found simultaneously in a single run.

GAs do not provide a single optimal solution for a problem, but several near-optimal solutions. This is the main advantage of evolutionary algorithms in the field of CAMD problems because near-optimal solutions can be further processed later by the designer and the most promising ones can be selected for synthesis.

During the operation of a GA, a population of candidate solutions competes for survival, based on their closeness to the target values. This closeness is described by a normalised distance value between 0 and 1 and called the *fitness*. The candidate molecules are usually described by the form of strings, and the components of these strings represent the 'genes' of the individual. Therefore, the evaluation of the population is carried out with the calculation of fitness, and the surviving members have the opportunity to reproduce and propagate their genes, thus forming the next generation. This propagation is dependent on the genetic operators applied by the specific algorithm being used; the most common ones are crossover and mutation. The creation of subsequent generations is continued until convergence is obtained (no considerable improvement is observed), or the maximum number of generations set by the user is achieved [15].

3 Description of the proposed algorithm

The developed algorithm, therefore, needs to solve effectively the CAMD tasks based on the needs of industrial and research work.

3.1 The utilised Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

The proposed structural multi-objective optimisation algorithm is based on the modification of the *fast non-dominated sorting approach* of NSGA-II [39, 40].

The randomly created initial parent population (with N members) is sorted based on non-domination. The crossover (Eq. (8)-(9)) and mutation (Eq. (10)-(11)) operators are applied to create the next generation (with N members) [41]. The algorithm applies an intermediate crossover, which creates two children from two parents: *parent1* and *parent2* (*child* and *parent* are vectors ($\boldsymbol{x}$) containing the results of the specific problems):

$$child1 = parent1 + rand \cdot ratio \cdot \left(parent2 - parent1 \right) \quad (8)$$

$$child2 = parent2 - rand \cdot ratio \cdot \left(parent2 - parent1 \right) \quad (9)$$

where the *ratio* is a scalar between 0 and 1. The applied Gaussian mutation adds a normally distributed random number to each variable:

$$child = parent + S \cdot randn \cdot \left(ub - lb \right) \quad (10)$$

$$S = scale \cdot \left(1 - shrink \cdot \frac{currGen}{maxGen} \right). \quad (11)$$

where *scale* is a scalar, that determines the standard deviation of the random number generated and *shrink* is a scalar between 0 and 1. As the optimisation progresses, this shrink parameter decreases the mutation range. *currGen* and *maxGen* are the numbers of the current and maximal generations, respectively.

Since elitism has been introduced, the creation of the first population differs from the creation of a subsequent one. The algorithm is described in terms of the t^{th} generation.

A combined population (R_t) (with $2N$ members) is created by the summation of the parent population (Pp_t) and the population obtained by the use of crossover and mutation operators (Q_t). The population R_t is sorted according to non-domination, and as all previous and current population members are included, elitism is ensured. Now solutions belonging to the first non-dominated front (F_1) are chosen for the next generation (Pp_{t+1}) (if the size of F_1 is smaller than N). This selection for the next generation is continued until the number of members from F_1 to F_i is larger than N. In these cases F_i is sorted based on the crowded-comparison operator, and the best solutions are chosen to fill the empty slots of the new population.

The defined input parameters are forwarded to a problem-specific genetic algorithm. This algorithm was built from the combination of a conventional NSGA-II algorithm with goal-oriented genetic operators with a method of correcting solutions called the *constraint correction*. The aim of modifications is to find feasible solutions more efficiently, which are difficult to find by conventional evolutionary algorithms due to the generation of a large number of impractical molecular structures.

3.2 Constraint Correction

The constraint correction algorithm ensures the feasibility of molecules by the stochastic modification of the originally impractical solution to a similar form of feasible molecular structure. The algorithm is presented in four sections for a better overview.

The algorithm starts with the triple bond correction part, as can be seen in Fig. 2. As was shown in Fig. 1 there are two types of groups containing triple bonds, $\boldsymbol{x}_9$ (≡X) and $\boldsymbol{x}_{10}$ (≡X–), so the correction of this part of the molecule means the rearrangement of these groups.

First the existence of triple bonds in the molecule is checked to decide if triple bond correction is needed or

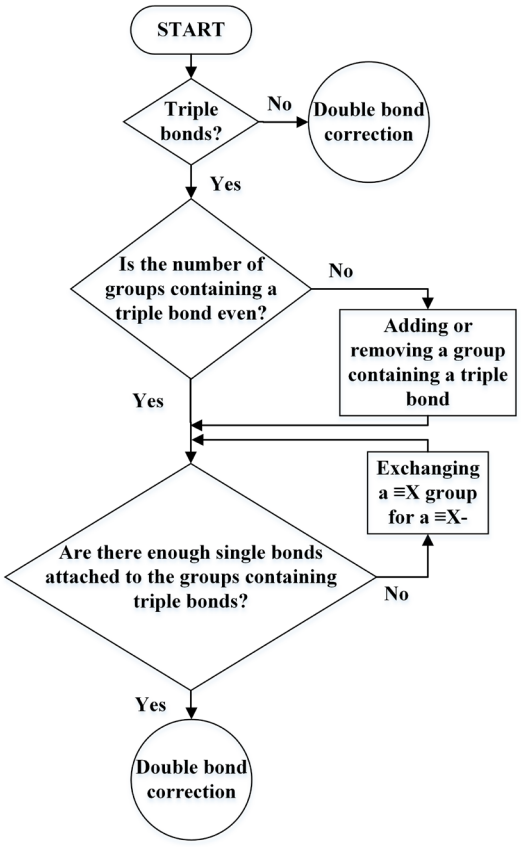

Fig. 2 The triple bond correction (Part I of constraint correction)

not. Then if the number of triple bonds is not even, the algorithm randomly adds or removes a group from the ≡X or ≡X– type of molecular fragments, resulting in feasible subunits containing random pairs of ≡X and ≡X– groups at this step of the algorithm. This means that, e.g. multiple X≡X pairs can exist separately in the structure, which is inadequate as a single molecular structure with properly connected fragments is required. The next step of the algorithm therefore replaces as many ≡X fragments with ≡X– fragments as needed for the connection of these subunits with the other parts of the molecule (this means that at least half of the groups containing triple bonds must be from the type of ≡X– fragments if the molecule is not X≡X). If the rearrangement of triple bonds is performed, the algorithm progresses to the double bond correction part presented in Fig. 3.

Analogous to the triple bond correction, the algorithm checks if the molecule contains double bonds for correction. In the case of double bonds molecular fragments of $\boldsymbol{\chi}_{5\text{-}8}$ (=X, =X–, =X< and =X=) are available, therefore the case of double bond correction is more complicated than triple bond correction. Since =X= can be available as well, a longer chain of groups connected with only double bonds can be formed. The existence of this chain is checked by the octet rule for double bonds as can be seen in Eq. (12):

$$\sum_{\substack{i \\ v_{double,i}>0}} \left(2 - v_{double,i}\right) x_i = 2, \tag{12}$$

where $v_{double,i}$ stands for the number of double bond corrections of the i^{th} molecular fragment (e.g. –X– has none; X=, =X– and =X< have one, while =X= has two). If the equation above is fulfilled, then a single, continuously connected structure (connected via only double bonds) can be formed from the molecular fragments containing double bonds. However, the molecular structure is practical only in terms of the double bonds. If the molecular structure contains other bonds as well the feasibility of the whole structure must be maintained, thus the rearrangement of double bonds may be required. First of all, if besides a =X= group no other groups with double bonds are contained in the structure, then the fragment is removed. As for the completion of this structure two other groups are needed, which is a bigger modification of the molecular structure, than the removal of only one =X= fragment. Then analogously to triple bonds, the number of fragments with double bonds is modified to be even by the addition or removal of one double-bond-fragment randomly (here the number of =X= groups is not considered, as the number of these fragments between two =X, =X– or =X< fragments is not limited). If the molecule contains only multiple =X= type double-bond fragments then the structure is completed with two =X, =X– or =X< fragments (to close the chain of fragments connected with only double bonds). At this step of the algorithm, molecular subunits connected with double bonds whose connections are needed to each other and to the other parts of the molecule through single-bond-units exist. The next step of the algorithm therefore replaces as many =X fragments with =X– or =X< fragments as needed for the connection of these subunits to the other parts of the molecule (this means at least half of the number of =X, =X– or =X< fragments must be of =X– or =X< type).

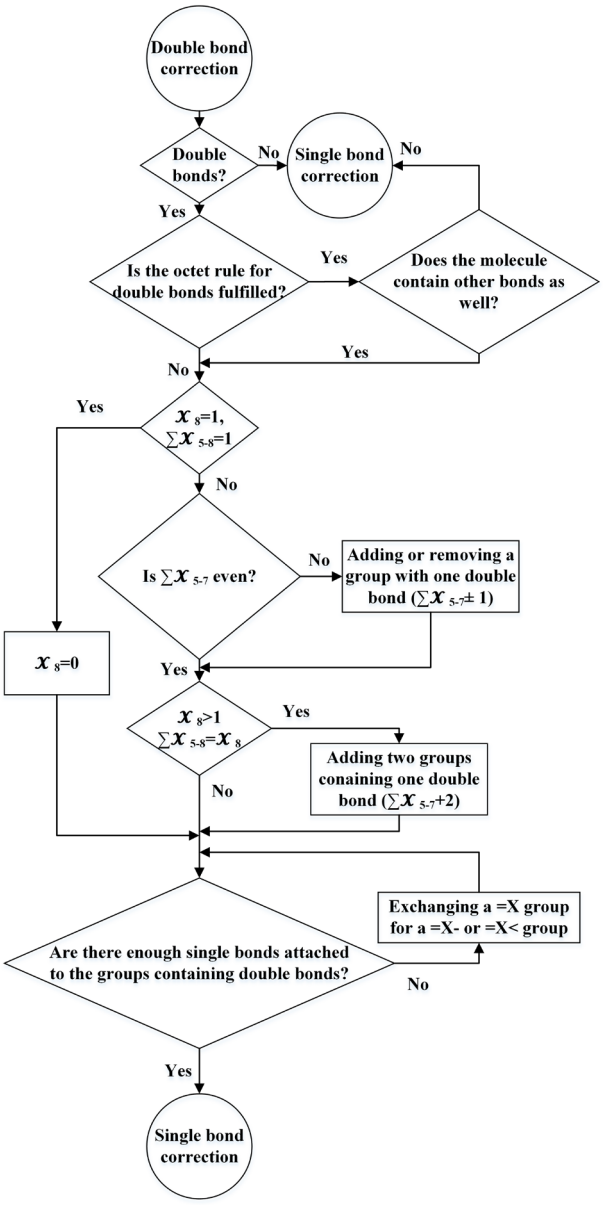

Fig. 3 The double bond correction (Part II of constraint correction) ($\boldsymbol{\chi}_5$ = |X=|, $\boldsymbol{\chi}_6$ = |=X–|, $\boldsymbol{\chi}_7$ = |=X<|, $\boldsymbol{\chi}_8$ = |=X=|)

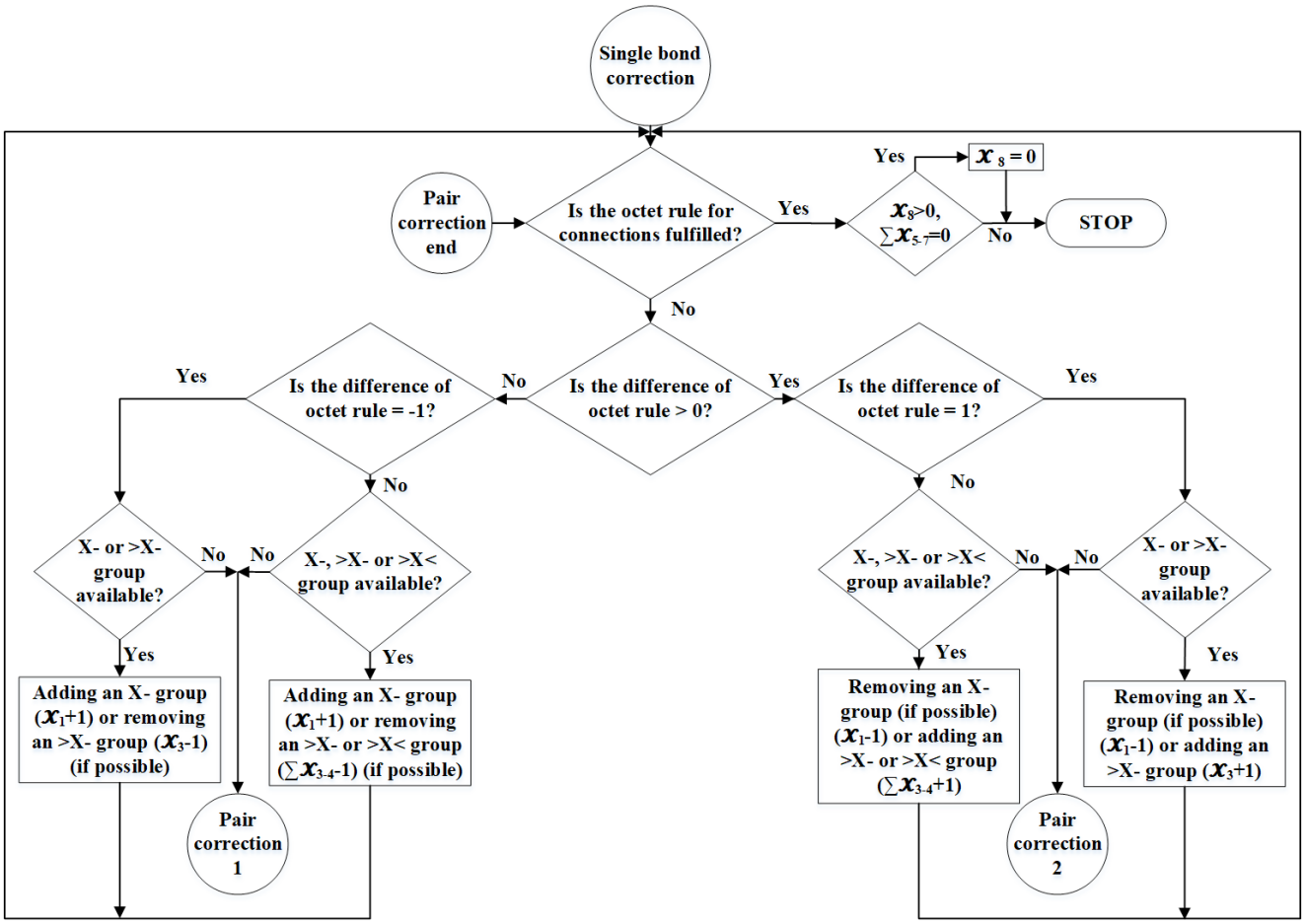

Fig. 4 The single bond correction (Part III of constraint correction) (x_1 = |X–|, x_2 = |–X–|, x_3 = |>X–|, x_4 = |>X<|, x_5 = |X=|, x_6 = |=X–|, x_7 = |=X<|, x_8 = |=X=|)

The next section of the algorithm is the single bond correction part as presented in Fig. 4. This part of the algorithm rearranges the structure to be feasible with the use of fragments having only single bonds. At the beginning of the algorithm double- and triple-bond-connected subunits and fragments with only single bonds (X–, –X–, –X<, >X<) can exist, which preferably would connect through fragments having only single bonds to form a continuously connected fragment-chain. To check if the connection of the subunits is fulfilled, the octet rule is applied for connections as presented in Eq. (3). If the octet rule for connections differs to the negative direction, the algorithm adds an X– fragment or removes an –X< or >X< fragment randomly, while if it differs to the positive direction, the algorithm removes an X– fragment or adds an –X< or >X< fragment randomly (after determining that the given operation can occur).

Sometimes in the case of complicated problems no fragments with only single bonds might be available as a variable. In this case, the pair correction section starts to change the number of double- or triple-bond-connected subunits in the optimal direction as can be seen in Fig. 5. If the octet rule differs to the negative direction, it adds a possible subunit, while if the octet rule differs to the negative direction it removes one. At this part of the algorithm, the randomly chosen modification is neglected, as this case does not occur in common problems. Stochasticity is ensured by the random modifications carried out by the NSGA-II algorithm.

Before stopping the algorithm, it is determined whether =X= type fragments were the only fragments with double bonds in the molecule due to the pair correction algorithm and =X= fragments are deleted if necessary.

3.3 Description of the tested algorithms

Four different versions of the proposed algorithm were implemented in MATLAB.

1. **The 'feasibility constraint' algorithm (Algorithm No. 1)**: The type of groups, their minimal and maximal numbers, the target properties and the property

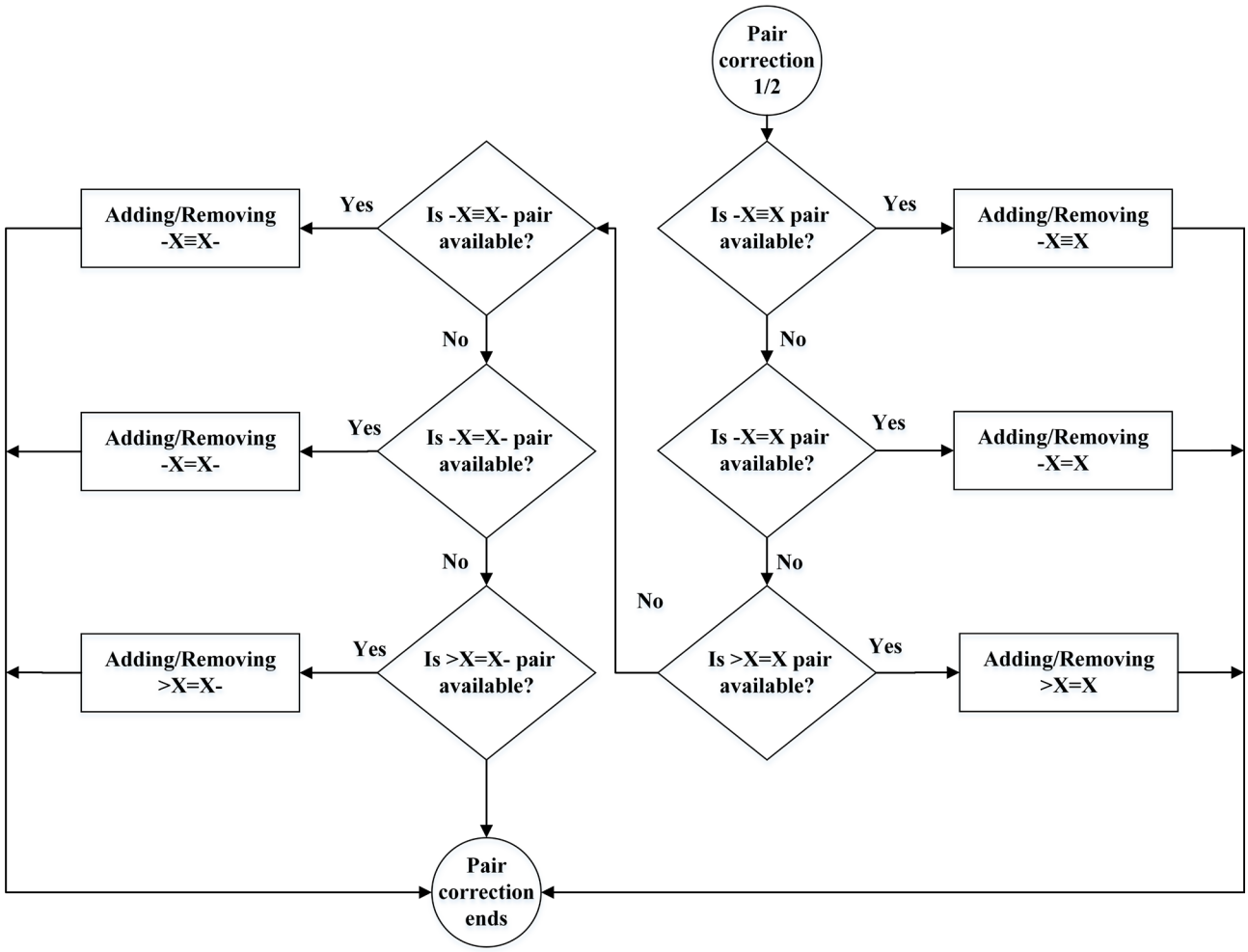

Fig. 5 The pair correction (Part IV of constraint correction)

constraints are defined. The original NSGA-II mutational and crossover operators are used. The intention of feasibility is maintained by the use of structural feasibility constraints described in Section 2 as hard constraints and by the application of constraint correction on the last population.

2. **The 'continuous constraint correction' algorithm (Algorithm No. 2):** Algorithm No. 1 is extended by constraint correction. The correction is carried out on not just the members of the last generation, but on all of the populations before the evaluation of fitness, inciting the reproduction of feasible molecular structures.
3. **The 'mutation rules' algorithm (Algorithm No. 3):** The mutation function of Algorithm No. 2 is changed, since the goal-oriented modification of the original "Gaussian" mutational algorithm of NSGA-II is applied in Algorithm No. 3. This is carried out in the interest of ensuring that the application of genetic operators cannot ruin the feasibility of the molecule. The application of original NSGA-II mutational operators is replaced by the use of predefined rules, and the use of crossover operators is neglected. During the mutation process, the algorithm chooses randomly from the predefined mutation rules and applies them to the parent structure. Based on eight simple principles 422 mutational rules were defined. These eight principles are:

- X– fragments can be replaced by X– groups (and vice versa)
- –X– fragments can be added or removed without restrictions
- –X=X– fragment pairs can be added or removed without restrictions (the possible preservation of =X= type groups is examined)
- >X–X– fragment pairs can be added or removed without restrictions
- =X– can be replaced by an =X< and an –X fragment (and vice versa)

- X– can be replaced by an X=X=X– fragment triple (and vice versa) (the possible preservation of =X= type groups is examined)
- X–, >X<, X– fragment triples can be added or removed without restrictions
- X– can be replaced by an X≡X– fragment pair (and vice versa)

During the definition of these principles, the most important aspect was to be able to modify the numbers of each fragment in the molecule, thus to include them in at least one principle and to ensure that by the application of any of these rules the structure stays feasible. During the use of the defined rules, the algorithm determines the possibility of application to avoid infeasible structures or negative variables in case of subtraction.

4. **The 'mutational rules with crossover' algorithm (Algorithm No. 4):** In the case of Algorithm No. 4 the mutational rules of Algorithm No. 3 are used, but in this case the crossover of NSGA-II ("Intermediate crossover") together with constraint correction is used as well to ensure the space of feasible solutions. The feasibility criterion as a hard constraint is still present.

The features of each applied algorithm are summarised in Table 1.

Table 1 The features of the applied algorithms

Feature	No. 1	No. 2	No. 3	No. 4
Feasibility as a hard constraint	✓	✓	✓	✓
Constraint correction of last population	✓	✗	✗	✗
Constraint correction of every population	✗	✓	✗	✗
Gaussian mutation	✓	✓	✗	✗
Mutational rules	✗	✗	✓	✓
Intermediate crossover (in the case of No. 4 with constraint correction)	✓	✓	✗	✓

4 The evaluation of the results

As the purpose of the current work is the development of an effective algorithm for the design of molecular structures to obtain the desired target properties, the comparison of the results is an essential task to determine the improvement and efficiency of the proposed formalisation against the results found in the literature. The efficiency inspection contains graphical and numerical methods as well. During the numerical evaluation, three efficiency indexes were determined:

- **Feasible solutions:** the number of non-dominated solutions of the last generation for each algorithm
- **Last changed population ($N_{gen,change}$):** the last generation in which the population of the Pareto-front is changed is also determined
- **Domination percentage (D_p):** the better performance of an algorithm is not just indicated from the larger number of solutions obtained by it, their domination compared to each other is important as well. For easier understanding, imagine the following didactic example. Algorithm *a*. has ten solutions, while the results of Algorithm *b.* give just nine solutions, from the ten members of the combined non-dominated fronts only 4 come from the results of Algorithm *a*. (D_p: 40%), and 6 come from Algorithm *b*. (D_p: 60%). This efficiency index is determined for both algorithms and compared to each other.

As the proposed genetic algorithm is a stochastic approach, ten runs of each algorithm were applied to evaluate the average efficiency, and the average of these indexes is presented in the Results section.

5 Results

To avoid the problem of random number generation of the stochastic method, the results of ten different runs were evaluated. The genetic algorithm worked for 250 generations with 100 population members in each. The effectivity of the different algorithms is compared through the following design tasks for the identification of different chemicals having the desired physical and chemical properties, estimated by the multidimensional property model. The results are downloadable from the website of the authors: www.abonyilab.com

5.1 Example I

The problem given in Friedler et al. [9] has been solved using the proposed genetic approach. The input parameters for the design task were as follows:

Available groups: $-CH_3$, $-CH_2-$, >CH-, >C<, –F and –Cl

Target properties: Max (T_b), Min (T_m)

Property constraints: $330\ K < T_b < 340\ K$

$130\ K < T_m < 140\ K$

The proposed algorithms found only one feasible structure having the given properties, as can be seen in Table 2

(the domination percentage is not given as no real Pareto-front was obtained). This result is similar to the results of Friedler et al. found only one feasible partition [9], this is the one found by us as well.

A possible structure with the given molecular fragments can be seen in Fig. 6.

5.2 Example II

This example is a modification of Example I as described by Friedler et al. [9]. The described problem is a good example to test the algorithms in the presence of single, double and triple bonds simultaneously.

Available groups: $-CH_3$, $-CH_2-$, >CH–, >C<, =CH2, =CH-, =C<, >C=O, ≡C−, –F, –Cl, –Br, –I, –O– and –OH

Target properties: Max (T_b), Min (T_m)

Property constraints: 330 K < T_b < 340 K

130 K < T_m < 140 K

Collecting all the resulted structures of the four algorithms during the ten runs we have found 26 separate structures. Friedler et al. found five feasible solutions and only one of the structures are common with the structures found by us. Dividing down this result to algorithms 4, 13, 8 and 23 different structures are found by the algorithms No. 1-4, respectively. According to Table 3 Algorithm No. 4 found the most solutions in average, thus the use of mutational rules and constraint correction to keep the optimisation in the space of feasible solutions and the use of crossover to keep the genetic diversity seems to be an efficient approach for the solution of the design task.

As can be seen in Table 4 Algorithm No. 4 did not just find the most solutions, but these solutions mainly dominate the solutions of other algorithms as well.

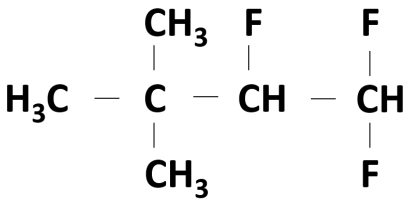

Fig. 6 A possible structure of the found fragments

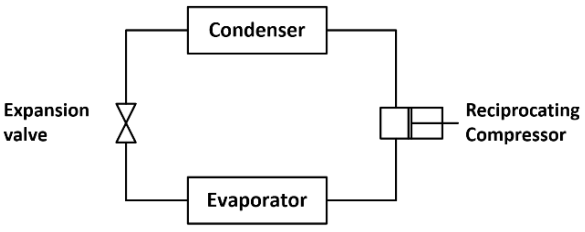

Fig. 7 A simple cooling circuit

Table 2 The results of Example I

# of algorithm	No 1	No 2	No 3	No 4
Feasible solutions	0.4	0.9	1.0	1.0
$N_{gen,change}$	34.7	21.8	60.7	63.1

Table 3 The results of Example II

# of algorithm	No 1	No 2	No 3	No 4
Feasible solutions	0.4	2.4	1.0	4.8
$N_{gen,change}$	53.4	105.5	206.2	246.2

5.3 Refrigerant design

During the operation of a cooling circuit, the circulated liquid must meet several requirements. In order to formalise the problem, imagine a simple cooling circuit with a condenser, a reciprocating compressor, an evaporator and an expansion valve, similarly as presented in Fig. 7.

The problem with the design of appropriate refrigerants described by Joback in [42] has been reexamined under the following conditions:

- P_{vp} (T = -1.1 °C) > 1.4 bar

The lowest pressure in the cycle should be greater than atmospheric pressure to reduce the possibility of air and moisture leaking into the system. The maximisation of P_{vp} (T = -1.1 °C) was defined as a target:

- P_{vp} (T = 43.3 °C) < 14 bar

High vapour pressure increases the size, weight and cost of the equipment. The minimisation of P_{vp} (T = 43.3 °C) was defined as a target:

- ΔH_v (T = −1.1 °C) > 18.4 kJ/(g·mol)

A larger enthalpy of vaporisation occurs in the smaller amount of used refrigerant. The maximisation of ΔH_v (T = −1.1 °C) was defined as a target:

- $C_{p,L}$ (T = 21.1 °C) < 32.2 cal/(g·mol·K)

Low liquid heat capacity prevents (or reduces the likelihood of) the refrigerant from flashing upon passage through the expansion valve. The heat capacity is evaluated at a constant temperature. The minimisation of $C_{p,L}$ (T = 21.1 °C) was defined as a target.

Besides the described targets, all the conditions were set as constraints.

Available groups: $-CH_3$, $-CH_2-$, >CH–, >C<, $=CH_2$, =CH–, =C<, =C=, ≡CH, ≡C–, –F, –Cl, –Br, –I, –OH, –O–, >C=O, O=CH–, –COOH, –COO–, =O, $-NH_2$, >NH, >N–, –CN, $-NO_2$, –SH, –S–

The results of refrigerant design can be seen in Table 5 and Table 6. As the value of $N_{gen,change}$ (last changed population) is quite high (250 is the maximum number of

Table 4 The domination percentage compared between pairs of algorithms (Example II)

Pareto set	No 1 [%]	No 2 [%]	No 3 [%]	No 4 [%]
No 1 [%]	40.00	30.00	36.67	14.60
No 2 [%]	100.00	60.00	89.00	40.32
No 3 [%]	56.67	28.50	50.00	25.48
No 4 [%]	93.42	62.78	83.45	50.00

Table 5 The results of refrigerant design

# of algorithm	No 1	No 2	No 3	No 4
Feasible solutions	20.5	84.3	99.5	99.7
$N_{gen,change}$	201.3	250.0	250.0	250.0

Table 6 The domination percentage compared between pairs of algorithms for the refrigerant design problem

Pareto set	No 1 [%]	No 2 [%]	No 3 [%]	No 4 [%]
No 1 [%]	15.00	13.47	12.18	12.13
No 2 [%]	86.53	50.00	45.86	45.81
No 3 [%]	87.82	54.14	50.00	49.95
No 4 [%]	87.87	54.19	50.05	50.00

generations), the algorithms seem to evolve effectively towards the Pareto front.

Most of the solutions were found by Algorithms No. 3 and 4, the number of members in the populations was almost 100. These solutions mainly dominate the solutions of Algorithms No. 1 and 2, as is represented by the domination percentage. This shows that the mutational rules alone or together with intermediate crossover (completed with constraint correction), and the restriction of genetic algorithms to the space of feasible molecules is an effective approach for solving the design problem.

The obtained results were compared to the results of Joback (the results were filtered according to the corresponding design conditions). According to Fig. 8 the presented method found significantly more structures (Joback found only 45 non-ring molecular structures) and the obtained results are better in terms of property values as well, however, there are negative values of liquid heat capacities. This result highlights an important remark on material design: all approaches are as good as they are allowed to be according to the used property estimation methods. The false results of negative liquid heat capacity values draw attention to the importance of the target and constraint definition and to the drawback that any proposed approach is highly determined by the uncertainty of the used property estimation methods.

To obtain physically interpretable results, a lower bound of 0 cal/molK was defined for liquid heat capacity, and the optimisation was repeated. The results can be seen in Table 7 and Table 8. Even though fewer structures have been found, the effectivity of the algorithms is still noticeable, as Algorithms No 2-4 found significantly more results than Joback.

Fig. 9 shows the properties of the obtained molecular structures with lower liquid heat capacity constraints

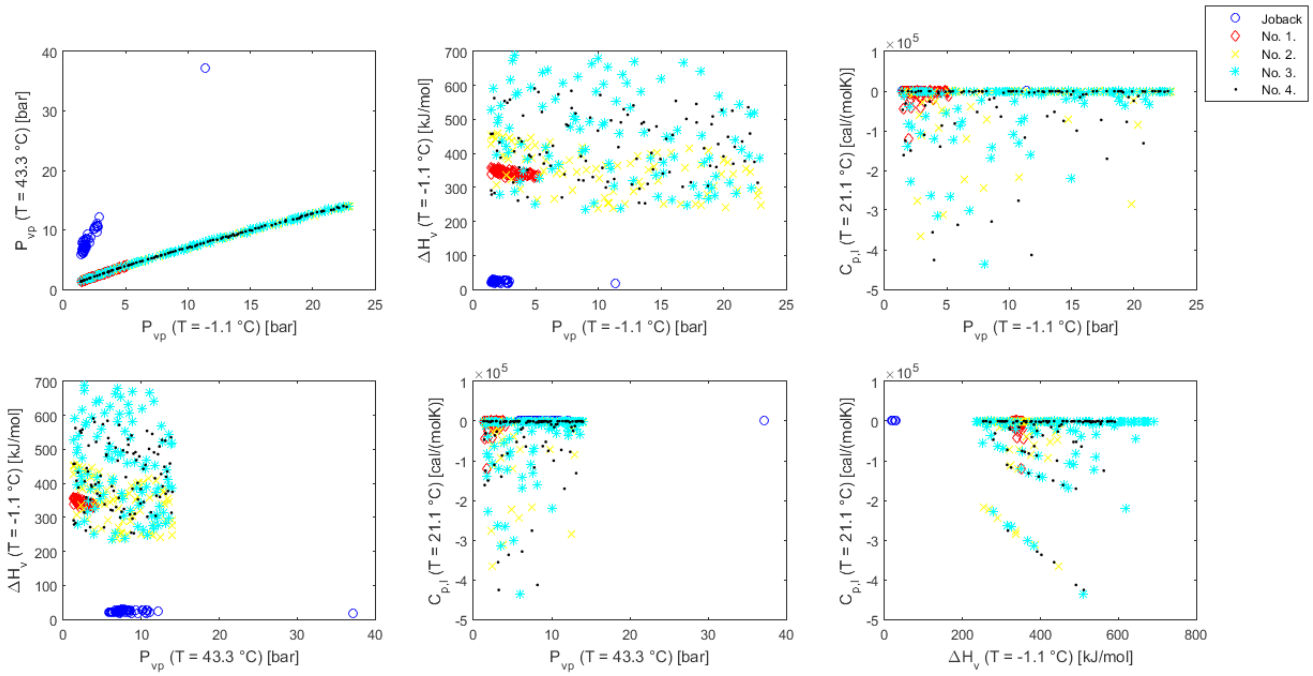

Fig. 8 Comparison between the properties of refrigerant structures found by the proposed methods and by Joback [42]

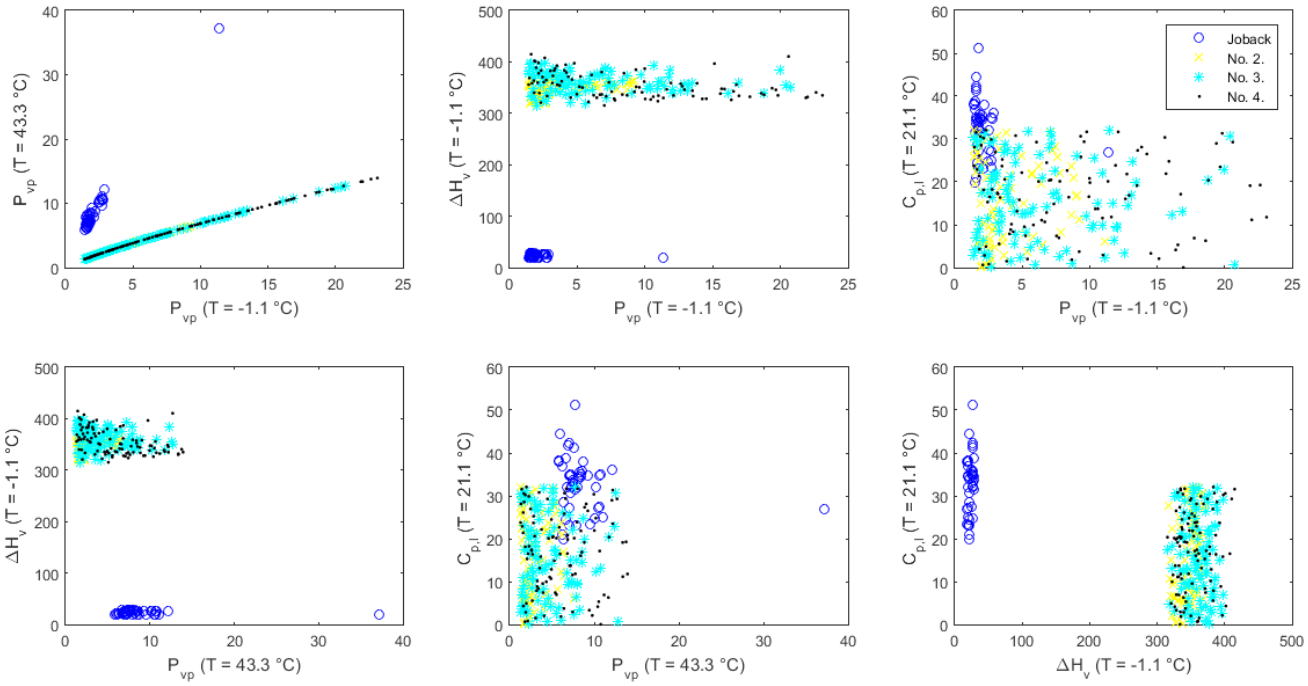

Fig. 9 Comparison of the results found by the proposed methods (with lower liquid heat capacity constraint) and by Joback [42]

Table 7 The results of refrigerant design with lower liquid heat capacity constraint

# of algorithm	No 1	No 2	No 3	No 4
Feasible solutions	0.0	57.0	60.8	74.1
$N_{gen,change}$	144.9	243.8	249.9	249.9

Table 8 The domination percentage compared between pairs of algorithms with lower liquid heat capacity constraints for the refrigerant design problem

Pareto set	No 1 [%]	No 2 [%]	No 3 [%]	No 4 [%]
No 1 [%]	0.00	0.00	0.00	0.00
No 2 [%]	90.00	45.00	49.92	38.23
No 3 [%]	100.00	50.48	55.00	43.16
No 4 [%]	100.00	61.77	57.24	50.00

compared to properties of structures found by Joback. According to the Figure, the properties of the found structures significantly outperform the structures found by Joback in the case of every target property.

6 Summary

The design of new molecules with specified properties is of increasing importance in the modern chemical industry and research. To effectively handle structural constraints we introduce goal oriented genetic operators to the multi-objective Non-dominated Sorting Genetic Algorithm-II (NSGA-II). The constraints are defined based on the hierarchical categorisation of the molecular fragments. The presented examples have demonstrated that the proposed genetic approach is capable of solving the formulated problems and moves effectively towards the Pareto-optimal front. With the multiple candidate molecules of the resultant Pareto front, several further targets can be taken into consideration: financial aspects, toxicity, availability, taste, aroma, colour, etc., thus the designer can consider personal insights during the design task.

The presented algorithms have a wider application range compared to the previously published material in [32]. The defined methods worked with varying degrees of effectivity in the test problems. From the results of Algorithm No. 1 it is obvious that the constraint correction of the resultant structures is far less effective compared to the goal-oriented modification of genetic operators as can be seen in Algorithms No. 2-4. The introduction of constraint correction in the field of the discrete optimisation problems of molecular design is a novel approach, as such methods are mainly applied in the case of continuous optimisation problems. The goal-oriented mutational operator tested in Algorithm No. 3 confines the variable vectors to the space of feasible solutions, thus exhibits an increased efficiency as illustrated by the examples. Algorithm No. 4 exhibits the combined potency of confining the variables to the space of feasibility via the application of goal-oriented mutational rules and the application of constraint correction, but to preserve genetic diversity intermediate crossover is applied as well.

The proposed approach is highly efficient to solve the molecular design problems as proven by the presented problems and ready for the design of hypothetical components in flowsheeting simulators. A possible future improvement of the algorithm can be the design of spatial structure of the molecule from the found fragments.

The proposed method was implemented in MATLAB. The results are reproducible since all the functions and numerical experiments are downloadable from the website of the authors: www.abonyilab.com

Annexes

Prediction of vapour pressure

The prediction of vapour pressure was carried out by the application of the Riedel-Plank-Miller equation-oriented technique. According to Eq. (13) the vapour pressure is a function of T_b, T_{br} and P_c values,

$$P_{vp} = P_{vp}\left(T_b, T_{br}, P_c\right). \tag{13}$$

The applied T_b, T_{br} and P_c values are estimated with the use of the Joback method. The vapour pressure can be calculated using Eq. (14)-(17).

$$\ln P_{vp,r} = -\frac{G}{T_r}\left[1 - T_r^{\,2} + k\left(3 + T_r\right)\left(1 - T_r\right)^3\right]. \tag{14}$$

$$G = 0.4835 + 0.4605h. \tag{15}$$

$$k = \frac{\frac{h}{G} - \left(1 + T_{br}\right)}{\left(3 + T_{br}\right)\left(1 - T_{br}\right)^2}. \tag{16}$$

$$h = T_{br}\frac{\ln\left(P_c/1.01325\right)}{1 - T_{br}}. \tag{17}$$

T_b and T_{br} are measured in K and P_c is measured in bars.

Prediction of enthalpy of vaporisation

The prediction of enthalpy of vaporisation was carried out with the use of the Watson relation as presented in Eq. (18). The ΔH_{vb} value is obtained from the Joback estimation method.

$$\Delta H_v = \Delta H_{vb}\left(\frac{T_c - T}{T_c - T_b}\right)^{0.38} \tag{18}$$

Prediction of liquid heat capacity

The prediction of liquid heat capacity is carried out with the use of the Rowlinson equation-oriented technique as a function of an acentric factor, ideal gas heat capacity and critical temperature, as seen in Eq. (19),

$$C_{pL} = C_{pL}\left(\omega, C_p^0, T_c\right). \tag{19}$$

C_p^0 and T_c values are estimated by the Joback method.

The acentric factor is estimated by the Lee-Kesler equation-oriented technique, as presented in Eq. (20).

$$\omega = \frac{-\ln P_c - 5.92714 + \frac{6.09648}{T_{br}} + 1.28862\ln T_{br} - 0.169347\cdot T_{br}^{\,6}}{15.2518 - \frac{15.6875}{T_{br}} - 13.4721\ln T_{br} + 0.43577\cdot T_{br}^{\,6}} \tag{20}$$

T_b and T_{br} are measured in K and P_c is measured in bars. Their values are obtained from the Joback estimation method.

Therefore, the Rowlinson equation-oriented technique uses Eq. (21).

$$\frac{C_{p,L} - C_p^0}{R} = 2.56 + 0.436\left(1 - T_r\right)^{-1} + \omega\left[2.91 + 4.28\left(1 - T_r\right)^{-1/3} T_r^{-1} + 0.296\left(1 - T_r\right)^{-1}\right] \tag{21}$$

List of Abbreviations

$C^R(K,N)$	selection of N groups from a set of K groups
$x_1, x_2, \ldots, x_n$	the number of group type #1, #2, …, #n, respectively
$\boldsymbol{x}$	n dimensional vector of positive integers (x_i) representing the variable vector
P_k	a specified property
f_k	property estimation function
$\boldsymbol{P}$	m dimensional vector of properties
$\boldsymbol{\chi}_{1-10}$	number of elements in the set of fragments (see Fig. 1)
"X–" , etc.	set of fragments (see Fig. 1)
$\widehat{P}_q$	target property value of target q
U_q	utility function of target q
$\boldsymbol{U}$	nq dimensional utility vector
Ω	space of feasible solutions
$g_s(\boldsymbol{x})$	set of structural equality constraints
$D_s(\boldsymbol{x})$	set of divisibility constraints
P_c^l	the lower bound of the specified property
P_c^u	the upper bound of the specified property
v_i	the connections of group i
P_{est}^k	estimated property value

θ_i^k	group contribution value of group i for property k
θ_0^k	offset value for property k
T_b	normal boiling point
T_m	normal melting point
T_c	critical temperature
P_c	critical pressure
V_c	critical volume
$H_{formation}$	heat of formation
$G_{formation}$	Gibbs free energy of formation
C_p^0	heat capacity
ΔH_{vap}	heat of vaporisation
ΔH_{fus}	heat of fusion
η_L	liquid dynamic viscosity
$parent1$, $parernt2$	vectors ($\boldsymbol{x}$) initially variables of crossover
$child1$, $child2$	vectors ($\boldsymbol{x}$) offspring variables of crossover
$ratio$	a scalar between 0 and 1
$rand$	a generated random number
$randn$	a generated n dimensional vector of random numbers
$currGen$	the number of the current generation
$maxGen$	the number of the maximal generation
$scale$	a scalar, that determines the standard deviation of the random number generated
$shrink$	a scalar between 0 and 1. As the optimisation progresses this shrink parameter decreases the mutational range
R_t	a combined population (with $2N$ members)
Pp_t	parent population
Q_t	the population obtained by the use of crossover and mutational operators
F	the non-dominated front
$v_{double,i}$	the number of double bond connections of the j^{th} molecular fragment
$N_{gen,change}$	the last generation in which the population of the Pareto front is changed
D_p	Domination percentage
P_{vp}	vapour pressure
ΔH_v	enthalpy of vaporisation
$C_{p,L}$	liquid heat capacity
ω	acentric factor

Acknowledgements

The research has been supported by the National Research, Development and Innovation Office – NKFIH, through the project OTKA – 116674 (Process mining and deep learning in the natural sciences and process development). Gyula Dörgő was supported by the ÚNKP-14-2-I New National Excellence Program of the Ministry of Human Capacities".

References

[1] Charpentier, J.-C. "Among the trends for a modern chemical engineering, the third paradigm: The time and length multiscale approach as an efficient tool for process intensification and product design and engineering", Chemical Engineering Research and Design, 88(3), pp. 248–254, 2010. https://doi.org/10.1016/j.cherd.2009.03.008

[2] Mao, Z., Yang, C. "Computational chemical engineering – Towards thorough understanding and precise application", Chinese Journal of Chemical Engineering, 24(8), pp. 945–951, 2016. https://doi.org/10.1016/j.cjche.2016.04.037

[3] Thienen, S. Van, Clinton, A., Mahto, M., Sniderman, B. "Industry 4.0 and the chemicals industry - Catalyzing transformation through operations improvement and business growth", Deloitte University Press, 2016. [online] Available at: https://www2.deloitte.com/insights/us/en/focus/industry-4-0/chemicals-industry-value-chain.html [Accessed: 18 December 2017]

[4] Yamamoto, H., Tochigi, K. "Computer-Aided Molecular Design to Select Foaming Agents Using a Neural Network Method", Industrial & Engineering Chemistry Research, 47(15), pp. 5152–5156, 2008. https://doi.org/10.1021/ie0712611

[5] Schneider, G., Wrede, P. "Artificial neural networks for computer-based molecular design", Progress in Biophysics and Molecular Biology, 70(3), pp. 175–222, 1998. https://doi.org/10.1016/S0079-6107(98)00026-1

[6] Manallack, D. T., Livingstone, D. J. "Neural networks in drug discovery: have they lived up to their promise?", European Journal of Medicinal Chemistry, 34(3), pp. 195–208, 1999. https://doi.org/10.1016/S0223-5234(99)80052-X

[7] Perez, R. E., Behdinan, K. "Particle swarm approach for structural design optimization", Computers & Structures, 85(19-20), pp. 1579–1588, 2007. https://doi.org/10.1016/j.compstruc.2006.10.013

[8] Hartenfeller, M., Proschak, E., Schüller, A., Schneider, G. "Concept of Combinatorial De Novo Design of Drug-like Molecules by Particle Swarm Optimization", Chemical Biology & Drug Design, 72(1), pp. 16–26, 2008. https://doi.org/10.1111/j.1747-0285.2008.00672.x

[9] Friedler, F., Fan, L. T., Kalotai, L., Dallos, A. "A combinatorial approach for generating candidate molecules with desired properties based on group contribution", Computers & Chemical Engineering, 22(6), pp. 809–817, 1998. https://doi.org/10.1016/S0098-1354(97)00253-6

[10] Holenda, B., Dallos, A., Nagy, Á., Friedler, F., Fan, L.-T. "A combinatorial approach for generating environmentally benign solvents and separation agents", Chemical Engineering Transactions, 3, pp. 871–875, 2003.

[11] Wang, Y., Achenie, L. E. K. "A hybrid global optimization approach for solvent design", Computers & Chemical Engineering, 26(10), pp. 1415–1425, 2002. https://doi.org/10.1016/S0098-1354(02)00118-7

[12] Mitra, K. "Genetic algorithms in polymeric material production, design, processing and other applications: a review", International Materials Reviews, 53(5), pp. 275–297, 2008. https://doi.org/10.1179/174328008X348174

[13] Weber, L. "Evolutionary combinatorial chemistry: application of genetic algorithms" Drug Discovery Today, 3(8), pp. 379–385. 1998. https://doi.org/10.1016/S1359-6446(98)01219-7

[14] Herring, R. H., Eden, M. R. "Evolutionary algorithm for de novo molecular design with multi-dimensional constraints", Computers & Chemical Engineering, 83, pp. 267–277, 2015. https://doi.org/10.1016/j.compchemeng.2015.06.012

[15] Venkatasubramanian, V., Sundaram, A., Chan, K., Caruthers, J. M. "11 − Computer-Aided Molecular Design Using Neural Networks and Genetic Algorithms", In: Devillers, J. (ed.) Genetic Algorithms in Molecular Modeling, Academic Press, London, 1996, pp. 271–302. https://doi.org/10.1016/B978-012213810-2/50012-8

[16] Kasat, R. B., Ray, A. K., Gupta, S. K. "Applications of Genetic Algorithm in Polymer Science and Engineering", Materials and Manufacturing Processes, 18(3), pp. 523–532, 2003. https://doi.org/10.1081/AMP-120022026

[17] Perdomo, F. A., Perdomo, L., Millán, B. M., Aragón, J. L. "Design and improvement of biodiesel fuels blends by optimization of their molecular structures and compositions", Chemical Engineering Research and Design, 92(8), pp. 1482–1494, 2014. https://doi.org/10.1016/j.cherd.2014.02.011

[18] Holland, J. H. "Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence", MIT Press, Cambridge, MA, USA, 1992.

[19] Banerjee, S., Rondoni, L., Mitra, M. "Applications of Chaos and Nonlinear Dynamics in Science and Engineering", Springer, Berlin Heidelberg, 2012. https://doi.org/10.1007/978-3-642-34017-8

[20] Wu, M.-S., Lin, Y.-L. "Genetic algorithm with a hybrid select mechanism for fractal image compression", Digital Signal Processing, 20(4), pp. 1150–1161, 2010. https://doi.org/10.1016/j.dsp.2009.12.009

[21] Michalewicz, Z. "Genetic algorithms + data structures = evolution programs", 3rd ed., Springer-Verlag, Berlin, Heidelberg, 1996. https://doi.org/10.1007/978-3-662-03315-9

[22] Renner, G., Ekárt, A. "Genetic algorithms in computer aided design", Computer-Aided Design, 35(8), pp. 709–726, 2003. https://doi.org/10.1016/S0010-4485(03)00003-4

[23] Deep, K., Thakur, M. "A new mutation operator for real coded genetic algorithms", Applied Mathematics and Computation, 193(1), pp. 211–230, 2007. https://doi.org/10.1016/j.amc.2007.03.046

[24] Deep, K., Thakur, M. "A new crossover operator for real coded genetic algorithms", Applied Mathematics and Computation, 188(1), pp. 895–911, 2007. https://doi.org/10.1016/j.amc.2006.10.047

[25] Colanzi, T. E., Vergilio, S. R. "A feature-driven crossover operator for multi-objective and evolutionary optimization of product line architectures", Journal of Systems and Software, 121, pp. 126–143, 2016. https://doi.org/10.1016/j.jss.2016.02.026

[26] Strug, B., Grabska, E., Ślusarczyk, G. "Supporting the design process with hypergraph genetic operators", Advanced Engineering Informatics, 28(1), pp. 11–27, 2014. https://doi.org/10.1016/j.aei.2013.10.002

[27] Salimi, R., Motameni, H., Omranpour, H. "Task scheduling using NSGA II with fuzzy adaptive operators for computational grids", Journal of Parallel and Distributed Computing, 74(5), pp. 2333–2350, 2014. https://doi.org/10.1016/j.jpdc.2014.01.006

[28] Venkatasubramanian, V., Chan, K., Caruthers, J. M. "Computer-aided molecular design using genetic algorithms", Computers & Chemical Engineering, 18(9), pp. 833–844, 1994. https://doi.org/10.1016/0098-1354(93)E0023-3

[29] Dyk, B. van, Nieuwoudt, I. "Computer aided molecular design of solvents for distillation processes", In: International Conference on Distillation & Absorption, Baden-Baden, Germany, 2002. [online] Available at: http://folk.ntnu.no/skoge/prost/proceedings/distillation02/dokument/1-1.pdf [Accessed: 10 December 2017]

[30] Glen, R. C., Payne, A. W. R. "A genetic algorithm for the automated generation of molecules within constraints", Journal of Computer-Aided Molecular Design, 9(2), pp. 181–202, 1995. https://doi.org/10.1007/bf00124408

[31] Brown, N., McKay, B., Gilardoni, F., Gasteiger, J. "A Graph-Based Genetic Algorithm and Its Application to the Multiobjective Evolution of Median Molecules", Journal of Chemical Information and Computer Sciences, 44(3), pp. 1079–1087, 2004. https://doi.org/10.1021/ci034290p

[32] Dörgő, G., Abonyi, J. "Group Contribution Method-based Multi-objective Evolutionary Molecular Design", In: Hungarian Journal of Industry and Chemistry, 44(1), pp. 39–49. 2016. https://doi.org/10.1515/hjic-2016-0005

[33] Poling, B. E., Prausnitz, J. M., O'Connell, J. P. "The properties of gases and liquids", 5th ed., McGraw-Hill, New York, USA, 2001.

[34] Fonseca, C. M., Fleming, P. J. "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization", In: Proceedings of the 5th International Conference on Genetic Algorithms, San Francisco, CA, USA, 1993, pp. 416–423.

[35] Liu, S., Papageorgiou, L. G. "Multiobjective optimisation of production, distribution and capacity planning of global supply chains in the process industry", Omega, 41(2), pp. 369–382, 2013. https://doi.org/10.1016/j.omega.2012.03.007

[36] Nicolaou, C. A., Brown, N. "Multi-objective optimization methods in drug design", Drug Discovery Today: Technologies, 10(3), pp. e427–e435, 2013. https://doi.org/10.1016/j.ddtec.2013.02.001

[37] Fonseca, C. M., Fleming, P. J. "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation", IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 28(1), pp. 26–37, 1998. https://doi.org/10.1109/3468.650319

[38] Odele, O., Macchietto, S. "Computer Aided Molecular Design: A Novel Method for Optimal Solvent Selection", Fluid Phase Equilibria, 82, pp. 47–54, 1993. https://doi.org/10.1016/0378-3812(93)87127-M

[39] Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. "A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II", In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Merelo, J., Schwefel, H.-P. (eds.) Parallel Problem Solving from Nature PPSN VI, Vol. 1917, Springer, Berlin, Heidelberg, 2000, pp. 849–858. https://doi.org/10.1007/3-540-45356-3_83

[40] Deb, K. "Multi-objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems", Evolutionary Computation, 7(3), pp. 205–230, 1999. https://doi.org/10.1162/evco.1999.7.3.205

[41] Song, L. "NGPM-A NSGA-II. Program in Matlab. User Manual. Version 1.4", 2011. [online] Available at: https://usermanual.wiki/Pdf/NGPM20manual20v14.1073441263 [Accessed: 10 January 2018]

[42] Joback, K. G. "Designing Molecules Possessing Desired Physical Property Values", PhD Thesis, Department of Chemical Engineering, Massachusetts Institute of Technology, 1989. [online] Available at: https://dspace.mit.edu/handle/1721.1/14191 [Accessed: 12 December 2017]