Looking for Alternatives: Optimization of Energy Supply Systems without Superstructure

Mike Preuss[1(✉)], Philip Voll[2], André Bardow[2], and Günter Rudolph[3]

[1] Chair of Information Systems and Statistics, ERCIS, WWU Münster, Münster, Germany
mike.preuss@uni-muenster.de
[2] Institute of Technical Thermodynamics, RWTH Aachen University, Schinkelstr. 8, 52062 Aachen, Germany
{philip.voll,andre.bardow}@ltt.rwth-aachen.de
[3] Computational Intelligence Group, TU Dortmund, Dortmund, Germany
guenter.rudolph@tu-dortmund.de

Abstract. We investigate different *evolutionary algorithm* (EA) variants for structural optimization of energy supply systems and compare them with a deterministic optimization approach. The evolutionary algorithms enable structural optimization avoiding to use an underlying superstructure model. As result of the optimization, we are interested in multiple good alternative designs, instead of the one single best solution only. This problem has three levels: On the top level, we need to fix a structure; based on that structure, we then have to select facility sizes; finally, given the structure and equipment sizing, on the bottom level, the equipment operation has to be specified to satisfy given energy demands. In the presented optimization approach, these three levels are addressed simultaneously. We compare EAs acting on the top level (the lower levels are treated by a *mixed-integer linear programming* (MILP) solver) against an MILP-only-approach and are highly interested in the ability of both methods to deliver multiple different solutions and the time required for performing this task.

Neither state-of-the-art EA for numerical optimization nor standard measures or visualizations are applicable to the problem. This lack of experience makes it difficult to understand why different EA variants perform as they do (e.g., for stating *how* different two structures are), we introduce a distance concept for structures. We therefore introduce a short code, and, based on this short code, a distance measure that is employed for a *multidimensional scaling* (MDS) based visualization. This is meant as first step towards a better understanding of the problem landscape. The algorithm comparison shows that deterministic optimization has advantages if we need to find the global optimum. In contrast, the presented EA variants reliably find multiple solutions very quickly if the required solution accuracy is relaxed. Furthermore, the proposed distance measure enables visualization revealing interesting problem properties.

A.I. Esparcia-Alcázar et al. (Eds.): EvoApplications 2014, LNCS 8602, pp. 177–188, 2014.
DOI: 10.1007/978-3-662-45523-4_15

1 Introduction

We address the problem of synthesizing energy supply systems with regard to time-dependent heating and cooling demands. This problem can be treated on several different scales from a single building to urban systems. In the present study, we focus on medium-scale problems, e.g., an industrial site or a university campus comprising of several, but not very many buildings. In this class of problems, the energy demands are distributed spatially, so that heating and cooling can be supplied in both centralized and distributed fashion. Of course, any mixture between these two extremes may be suitable – as is usually the case.

Energy supply systems incorporate energy conversion plants (e.g., boilers), energy distribution infrastructure (e.g., heating pipelines and power cables), and energy storages. The synthesis of these integrated systems is a complex problem that has to be considered on three levels [2] (Fig. 1): on the top level, the *synthesis* level, the structure or configuration of the energy system is fixed; on the intermediate level, the *design* level, the technical specifications of the employed technical components have to be specified (e.g., nominal capacities and operating limits); finally, on the bottom level, the *operation* level, technical components' operation modes need to be specified for each instant of time. The three decision levels directly influence each other, and thus, for optimal synthesis, all three levels must be considered simultaneously.

For the optimization-based synthesis of energy supply systems, most commonly superstructure-based optimization methods are employed [5].The general superstructure optimization problem for energy supply systems synthesis is given by a mixed-integer nonlinear programming (MINLP) problem:

$$\min_{s,d,o} f(s,d,o), \quad \text{s.t. } h(s,d,o) = 0, \;\; g(s,d,o) \leq 0, \quad s \in S, d \in D, o \in O \qquad (1)$$

where the values of the decision variable vectors s, d, and o must be determined to minimize the objective function f. The decision variables are part of the continuous and/or integer variables space S, D, and O, which represent the synthesis (i.e., (non-)existence of a unit), design (i.e., unit sizing, etc.), and

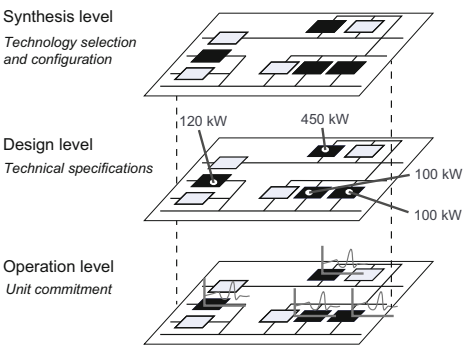

Fig. 1. Hierarchically-structured problem of energy systems synthesis on three levels

operation (i.e., flow rates, on/off-status of a unit, etc.) decision variable spaces, respectively.

It is crucial to understand that the designer has to decide *a priori* which alternatives should be included in the superstructure: On the one hand, the designer thereby runs the risk to exclude the optimal solution from consideration; on the other hand, to circumvent this problem, excessively large superstructures can be employed, which however lead to prohibitive computational effort for the solution of the optimization problem [4].

To avoid these issues, recently two methods have been proposed for the automated optimization-based synthesis of energy supply systems; an automated superstructure-based synthesis methodology [8] and an automated superstructure-free synthesis methodology [7]:

a) The superstructure-based synthesis methodology employs algorithms for automated superstructure generation and deterministic optimization. To find the optimal solution of a synthesis problem, this methodology performs successive superstructure expansion and optimization to continuously increase the number of units embedded in the superstructure until the final superstructure incorporating the optimal solution is found.
b) The superstructure-free methodology simultaneously generates and optimizes candidate solutions in search for the optimal solution. The methodology is based on a knowledge-integrated evolutionary algorithm that applies a handful of generic replacement rules for the evolution of solution structures.

In this work, linearized MILP formulation is employed for synthesis of energy supply systems [7]. For reasonably small test cases, synthesis problems can then be solved exactly in seconds or minutes, but for large-scale problems, the solution can take up to hours. However, if the structure is fixed – as is the case for the candidate solutions arising in the superstructure-free approach – the underlying design and operation problems can usually be solved as an MILP in a matter of seconds. In case of the superstructure-free approach, the problem is not solved exactly, however, it might be faster to find a very good solution heuristically than to wait for the optimum generated through deterministic search. But the main asset of the metaheuristic search is that we obtain several good solutions in one run. This is a major benefit for real-world planning problems because a single solution has only limited significance, , and thus decision makers usually prefer to obtain several promising alternatives that can be further evaluated with regard to further constraints arising in practice (e.g., changing constraints such as energy tariffs and energy demands).

The main task of this work is to investigate under which conditions a metaheuristic has advantages when compared to exact optimization algorithms for the type of structural problems we are dealing with, especially if several alternative solutions are desired. Therefore, first, the test case is described in detail in §2. We will experimentally compare an exact solution and the metaheuristic optimization in §6. However, we start with describing the superstructure-free synthesis methodology in §3. In order to quickly recognize the produced structures and be able to compute a distance between possible alternatives, we define

a shortcode, and based on that, a distance measure between structures in §4. Being equipped with a distance matrix, we can establish a *multidimensional scaling* (MDS) based visualization of our non-numerical search space in order to get a first idea of difficulties this problem contains and use this in order to select suitable optimization techniques. In §5, the different employed EA approaches are introduced and the different ways to solve an MILP by means of a solver alone (based on an successively extended superstructure) or in combination with an EA are explained.

2 Test Case

The test case represents a real-world problem from the pharmaceutical industry. The test case has already been analyzed in detail in [7]. The analyzed site consists of six building complexes housing offices, production and research facilities (Fig. 2). A public road separates the considered site into main site (A) and secondary site (B). On site A, all building complexes are connected by a central heating and cooling network. In the base case, site B is not connected to the cooling network, but only to the heating network. The connection of site B to the cooling network on site A is not allowed due the public road. Both sites are connected to the regional natural gas grid (gas tariff: 6 ct/kWh) and the regional electricity grid (electricity tariff: 16 ct/kWh; feed-in tariff: 10 ct/kWh). Electricity generated by the combined heat and power (CHP) engines can be used on-site to meet electricity demands or to run compression chillers, or else it can be fed to the regional electricity grid. All heat generators have to be installed on site A.

The described site has time-varying demands for heating, cooling, and electricity. modeled by monthly-averaged demand time series. The annual demands for electricity, heating, and cooling amount to 47.7 GWh, 28.1 GWh, and 27.3 GWh, respectively. The demand profiles are symmetric around the summer months July

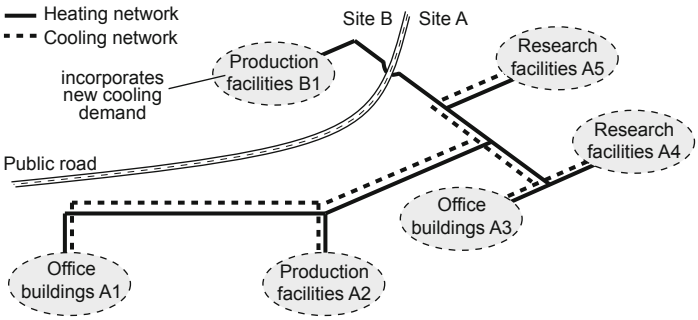

Fig. 2. Schematic plant layout of the considered site. On site A (main site), a central heating and cooling network connects five building complexes. The building complex on site B (secondary site) is only connected to the central heating network. Establishing new connections between both sites is impossible due to a separating public road. [7]

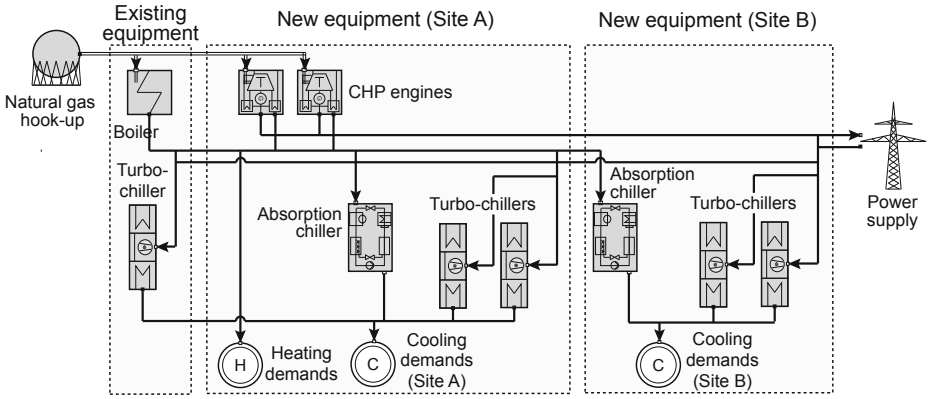

Fig. 3. Optimal flowsheet of the real-world synthesis problem. For simplicity, the electricity demand is not shown in the figure. [7]

and August. Thus, they are further simplified by aggregation to only six time steps.In addition, the minimum and maximum demands are taken into account. These demands occur only during few hours per year, however, it is important to incorporate them in the demand profiles to guarantee adequate equipment sizing. In total, the energy demands are modeled by eight time steps including the peak-load time steps.

The existing supply system consists of three boilers, one CHP engine, and three compression chillers. However, one boiler and one compression chiller cannot be further operated, and thus require substitution. Next to the given component types, we will also consider absorption coolers.

The optimal solution installs existing as well as new equipment. The optimal net present value adds up to $-46.99 \cdot 10^6$ EUR (Table 1) improving the base case by 39 %.

Table 1. Economic parameters of base case and NPV-optimal solution [7]

solution	NPV / 10^6 EUR	investments / 10^6 EUR	energy cost / 10^6 EUR p.a.	maintenance cost / 10^6 EUR p.a.
base case solution	−76.36	0	11.27	0.11
NPV-optimal solution	−46.99	2.35	6.44	0.22

3 Superstructure-Free Synthesis Methodology

The superstructure-free synthesis methodology proposed by [8] employs a hybrid optimization algorithm combining metaheuristic with deterministic optimization [6]. Metaheuristic optimization is realized by an evolutionary algorithm employing a mutation operator that randomly replaces substructure from a candidate solution by alternative structures. This approach allows for simultaneous

alternatives generation (on the synthesis level) and optimization (on the design and operation levels). The mutation operated is based upon a hierarchically-structured graph, the so-called *energy conversion hierarchy* (ECH) that classifies the considered energy conversion units according to their functions. This enables an efficient definition of all reasonable connections between the regarded technologies. Thus, a minimal set of generic replacement rules is then sufficient to employ structural mutation for the generation of any solution structure. For more details on this concept, the reader is kindly referred to [8].

The general mathematical programming problem for single-objective optimization based synthesis of energy supply systems is given by (1). Here, the decision variable vectors s, d, and o are part of the continuous and/or integer variable spaces S, D, and O, which represent the synthesis, design, and operation decision variable spaces, respectively. The three synthesis levels feature an inherent hierarchical structure, and thus the mathematical programming formulation can be decomposed into an upper level dealing with the synthesis, and a lower level dealing with the design and operation. Thus, the mathematical programming formulation can be reformulated as

$$\min_{s} \quad \hat{f}(s), \quad \text{s.t.} \min_{d,\,o} \; f^{(s)}(d, o).$$

Instead of explicitly modeling structural decisions in a superstructure, the presented mutation operator is embedded in an evolutionary algorithm that continuously evolves new configuration alternatives to perform optimization on the synthesis level. For equipment sizing and operation, rigorous MILP optimization is used as local refinement strategy; i.e., for each configuration alternative generated by mutation, an MILP problem is solved to identify the optimal equipment sizing and operation that maximizes the net present value. With net present value $C_{t_{\mathrm{CF}}}$ as objective function, the problem formulation of the hybrid optimization is given by

$$\max_{\sigma} \; \hat{C}_{t_{\mathrm{CF}}}(\sigma), \qquad \sigma \in \Sigma, \quad \text{s.t.} \max_{d,o} \; C^{(\sigma)}_{t_{\mathrm{CF}}}(d, o), \tag{2}$$

where σ represents a structure evolved by mutation, and Σ represents the set of all possible structures.

In this paper, the hybrid optimization is based on the MILP formulation presented by [7]. However, it should be noted again that the generic component-based modeling enables to use any other programming formulation as well.

4 Shortcode and Distance Measure

To simplify recognition of structures contained in actually evaluated solutions, a shortcode is defined that provides the types and numbers of the employed energy conversion plants. Note that the topology is omitted from this notation, so that it is possible that two solutions appear to be identical but have different topologies and thus different target values. The four different technology types boiler,

absorption chiller, compression chiller, and combined heat and power (CHP) engine are matched to the tokens `Bo, AC, CC`, and `CE`, followed by the number of plants per type encoded in a structure. As an example, `Bo1CC1` represents a structure that embeds one boiler and one compression chiller.

On base of this short code, we define a distance function over the different structures to obtain a numerical value. The function is given in (3) and resembles the euclidean distance with each of the four types (in alphabetical order, `AC`=1, `Bo`=2, `CC`=3, and `CE`=4), with N_{i1} and N_{i2} denoting the two structures. Two structures that embed a certain type of technology or not are considered more diverse than two structures that incorporate the same types of technologies but in different numbers. Thus, the second term in (3) with the signum function makes sure that the distance of two structures containing 0 and 1 units of a specific technology type are considered larger than for 1 and 2 or higher unit numbers.

$$dist_1(N_1, N_2) = \sqrt{\sum_{i=1}^{4}(|N_{i1} - N_{i2}| + \operatorname{sgn}(|N_{i1} - N_{i2}|))^2} \tag{3}$$

For so-called retrofit optimization, where a number of plants is already installed, it is necessary to add means that reveal if a plant is new or retained from the base case. We express the difference in the shortcode by writing existing plants with small letters, such that `AC1ac1bo2` denotes one new and one existing absorption chiller and two boilers. The distance function is adjusted appropriately in (4) with the introduction of n_1 und n_2 for the existing plants. The correction factor r (set to 2) in the last term connects old and new plants of the same types by adding the sum of these as additional 'dimension'.

$$dist_2(N_1, N_2, n_1, n_2) := \Big(\sum_{i=1}^{4}(|N_{i1} - N_{i2}| + \operatorname{sgn}(|N_{i1} - N_{i2}|))^2 + (|n_{i1} - n_{i2}| + \operatorname{sgn}(|n_{i1} - n_{i2}|))^2 + r(|N_{i1} + n_{i1} - N_{i2} - n_{i2}|)^2 \Big)^{1/2} \tag{4}$$

We obtain figure 4 by computing a distance matrix from 100 randomly chosen solutions by means of $dist_2$ and then using multidimensional scaling (MDS) as dimension reduction technique in order to map it into a 2-dimensional space. The best solutions are found in the middle, on the border to several invalid regions. Note that invalid solutions have the same objective function values, and thus evalution of these solutions provides no information for the optimization method on the search direction to reach an area of valid solutions. However, the chosen distance function appears to be meaningful because the resulting topology looks intuitive (as expected, similar structures are mapped to the same region of the target area).

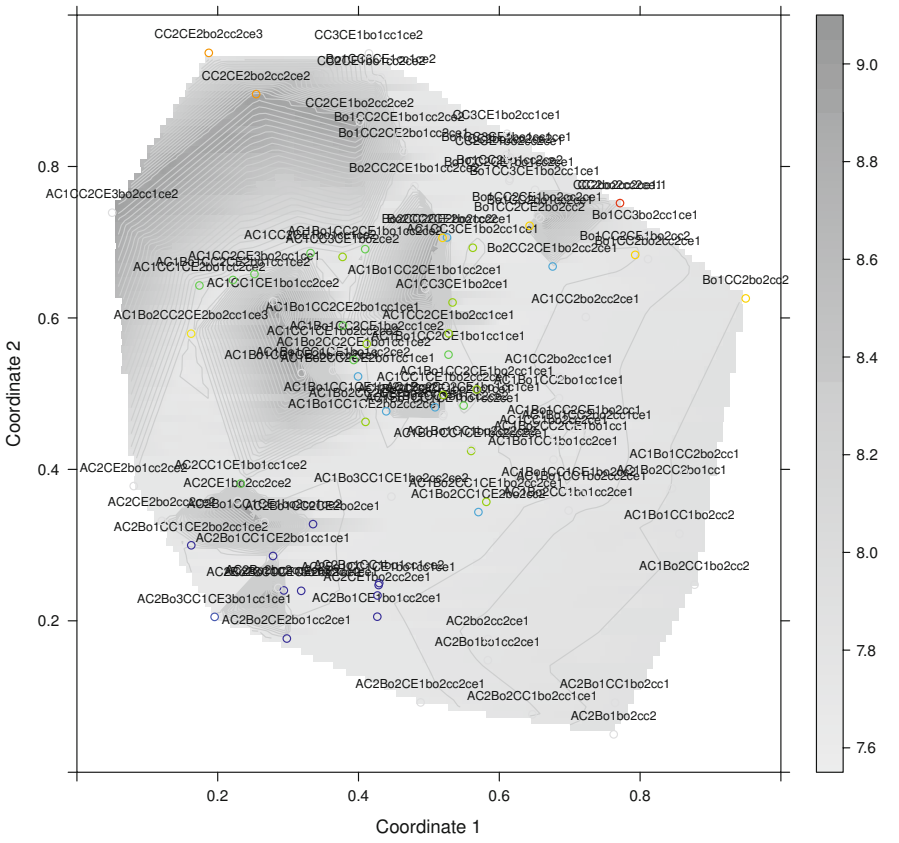

Fig. 4. Multidimensional scaling (MDS) based visualizaton of a random sample of size 100. The contour reveals the (log10 transformed) objective function values (NPV) of the different solutions, with invalid ones having a cost of €10^9.

5 MILP Solving and Evolutionary Approaches

As explained in §2, we are dealing with a 3-level hierarchical problem that may be approached in two very different ways: 1) by means of an MILP solver that solves a series of successively extended superstructure-based optimization problems (in the following referred to as the *purely MILP-based approach*) to return the exact global optimum (in case it can be solved) – however, depending on the problem size at considerably computational cost, i.e. long solution times; and 2), by means of a superstructure-free EA (in the following referred to as the *mixed approach*) that works on the top synthesis level of the optimization problem and uses an MILP solver to determine the solutions for the underlying design and operation levels. In both cases, the necessary computing times are usually much smaller for infeasible solutions, however, the computing times for feasible solutions can vary significantly due to the different complexities of the underlying design and

operation problems. From sensitivity analyses concerning result stability under shifts in demand data (introducing new time steps or changing values of existing time steps), we know that the optimal solution can easily vary by up to 2%. For this reason, we do not necessarily have to find the global optimal solution, but we strive for solutions with at most 1% deviation from the global optimum. In case of the combined approach, the same accuracy is required for the lower level MILP optimization.

As MILP solver, we employ SCIP [1], version 3.0.0, one of the fastest available non-commercial solvers. Note that exchanging the solver with a quicker commercial solver will reduce computing times for both approaches approximately by a factor of 10, according to our tests.

In our first tests with the mixed approach, we found that a lot of precious running time is lost by re-evaluating already considered solutions. Therefore, a tabu search-like [3] list of forbidden structures is implemented for all metaheuristics to follow. During the algorithm run, we keep track of the shortcodes for evaluated solutions. New solution candidates are produced by applying the mutation operator described in §3. However, they are only evaluated if they are either not yet contained in the list, or if 10^3 successive attempts fail to obtain an untested structure. Note that the topologies of solution candidates are not regarded, and multiple topologies may map to the same shortcode. It thus makes sense to allow the evaluation of a candidate with an already recorded shortcode as it may have a different topology. However, at least at the beginning of a run, this rarely happens because many different plant combinations are available.

In order to roughly estimate the size of the set of different structures (neglecting differences in the topology), we first consider the choice of already existing plants. We can choose any combination of 0 to 2 boilers, one or none CHP engine, and 0 to 2 absorption chillers, leading to $3 \cdot 2 \cdot 3 = 18$ possibilities. Let us assume that for each of these, we can add up to 10 new plants of $4 + 1$ types (`AC`, `Bo`, `CC`, `CE`, and *none*). Drawing 10 times from this set with replacement and without considering order results in $\frac{(n+k-1)!}{(n-1)!k!} = \frac{(5+10-1)!}{(5-1)!10!} = 1001$ possibilities for the added plants. This results in $18 \cdot 1001 - 1 = 18017$ type combinations without taking the topology into account. However, this is only a rough estimate because we allowed for a greater number of new plants, but this was only rarely realized during our relatively short runs as it requires a high number of successive mutations into one direction.

As algorithm types, we consider random search, random walk (implemented as (1,1)-EA), a (10+10)-EA and a (50+10)-EA, each of these utilizing a tabu list as described above. The reasoning behind using a population was to enable more parallelized search. Our EA employs an evolution strategy (ES) type selection, structural mutation as described above in §3, and no recombination.

6 Experimental Comparison

The two goals of the algorithm comparison are to find out, a) which metaheuristic-based approach reliably detects at least one near-optimal solution (objective value

$\leq 1\%$ from global optimum) faster than the integrated MILP-based approach, and b) which of the approaches can be recommended concerning the number of good alternatives it produces quickly.

Pre-experimental planning. During first tests, we found that, in most cases, 30 seconds suffice to solve the design and operation problem for a given structure. Therefore, the maximum time for this solution phase is constrained to 30 seconds. This means that the concerned evaluations return an objective value that is worse than it will be if the underlying optimization problem is solved to global optimality. On the other hand, we save precious time as the overall run length should be less than one day (24 h).

An additional test with a (1+1)-EA revealed that it usually gets stuck very early, and thus is mostly not able to reach the desired objective function value level. This may be surprising because the (1+1)-EA was allowed to perform restarts. However, it can be explained with the relatively short run length that did not enable more than a small number of restarts. This variant is therefore disregarded in the following.

Setup. The purely MILP-based (deterministic) approach is run until the desired accuracy of 1% is reached; the corresponding solution time is recorded. The 4 mixed approaches are run 10 times until 3000 evaluations have been spent. Note that the actual computing time for this is limited by $0.5\,\text{minutes} \cdot 3000 = 1500$ minutes. However, the true computing time varies between runs and usually takes about 60% of this value (the time consumed by the underlying MILP-solving cannot be predicted). The average number of mutations is set to 1.5 for the random walk and population-based EAs.

Task. A mixed approach is considered reliable only if it produces a solution within the 1% bound before reaching the time spent by the purely MILP-based approach in *every* run. We consider one metaheuristic better than another if it consistently provides more satisfactory solutions within a smaller average time.

Results/Visualization. For the given problem, the purely MILP-based approach via SCIP needs 619 minutes to reach the 1% bound. This is depicted as red line in the diagrams for the mixed approaches in fig. 5. Each row of the plot represents one of the 10 runs, and blue dots each stand for one (structurally different) solution with satisfactory quality. The number on the bottom right corner of each plot denotes the average number of satisfactory solutions obtained over the runs, at the top right corner the average time for reaching a satisfactory solution is depicted.

Observations. Random search generates only few satisfactory solutions, whose generation is not even necessarily faster than the solution provided by the purely MILP-based approach. Random walk, (10+10)-EA and (50+10)-EA produce many near-optimal solutions, however on average at proportionally larger computation times. We would like to add that during the runs, technical problems with SCIP were observed because it sometimes (in about 1 of 500 cases) crashed during the design and operation level optimization.

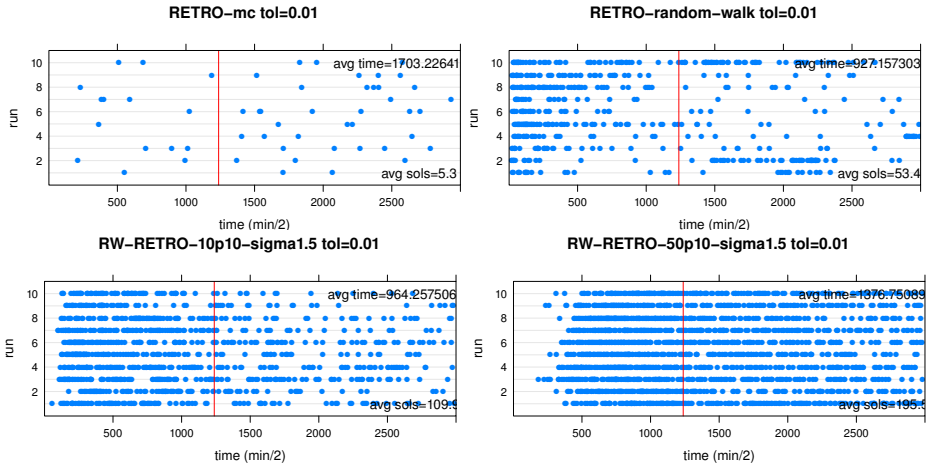

Fig. 5. Performance comparison of different random search and EA variants (all with tabu list), from left to right and top to bottom: random search, random walk, (10+10)-EA and (50+10)-EA. The red line represents the required time of the purely MILP-based approach to reach the desired accuracy. The blue dots each represent one satisfactory solution provided by the mixed approach.

Discussion. From the results we deduce that random search is obviously not the method of choice, as it is unreliable and does not generate many satisfactory solutions. The other three algorithms each have different strenghts: the (tabu list enhanced) random walk provides good solutions very quickly, but obtains much fewer of them if compared to the (10+10)-EA and the (50+10)-EA. With the two criteria given above, it is not possible to take a decision between them, they are uncomparable. If only response time is considered, the (tabu list) random walk appears best, if more solutions are needed, the slightly slower (10+10)-EA is recommended.

7 Conclusions

We compare several EA variants that employ an underlying MILP solver in order to solve a structural optimization problem without using superstructure models to a MILP-only approach that solves a series of successively extended superstructure models. The latter may have an advantage if we need to find the exact global optimum, while some of the proposed tabu-list enhanced EA variants reliably find multiple solutions very quickly if the required accuracy is relaxed a bit. Furthermore, our distance measure enables a visualization that reveals interesting problem properties. This should be helpful for improving the optimization process in the future. Additionally, we need to carefully analyze the distribution of the obtained solutions.

References

1. Achterberg, T.: SCIP: solving constraint integer programs. Mathematical Programming Computation **1**(1), 1–41 (2009)
2. Frangopoulos, C.A., von Spakovsky, M.R., Sciubba, E.: A brief review of methods for the design and synthesis optimization of energy systems. Int. J. Appl. Therm. **5**(4), 151–160 (2002)
3. Glover, F.: Future paths for integer programming and links to artificial intelligence. Comput. Oper. Res. **13**(5), 533–549 (1986)
4. Kallrath, J.: Mixed integer optimization in the chemical process industry: Experience, potential and future perspectives. Chem. Eng. Res. Des. **78**(6), 809–822 (2000)
5. Liu, P., Georgiadis, M.C., Pistikopoulos, E.N.: Advances in energy systems engineering. Ind. Eng. Chem. Res. **50**(9), 4915–4926 (2011)
6. Puchinger, J., Raidl, G.R.: Combining Metaheuristics and Exact Algorithms in Combinatorial Optimization: A Survey and Classification. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 41–53. Springer, Heidelberg (2005)
7. Voll, P., Klaffke, C., Hennen, M., Bardow, A.: Automated superstructure-based synthesis and optimization of distributed energy supply systems. Energy **50**, 374–388 (2013)
8. Voll, P., Lampe, M., Wrobel, G., Bardow, A.: Superstructure-free synthesis and optimization of distributed industrial energy supply systems. Energy **45**(1), 424–435 (2012)