©20xx IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

This is self-archiving version of an accepted article. The final version is here: 10.1109/CEC45853.2021.9504924

Compressor Schedule Optimization for a Refrigerated Warehouse Using Metaheuristic Algorithms

Rafał Biedrzycki
Warsaw University of Technology
Institute of Computer Science
Warsaw, Poland
rbiedrzy@elka.pw.edu.pl

Kamil Kwiatkowski
Euros Energy
Warsaw, Poland
kkwiatkowski@eurosenergy.com

Paweł Cichosz
Warsaw University of Technology
Institute of Computer Science
Warsaw, Poland
pcichosz@elka.pw.edu.pl

***Abstract*—This paper investigates the suitability of several metaheuristic algorithms for the problem of compressor schedule optimization for a refrigerated warehouse. A realistic simulator of such a warehouse is used, based on domain knowledge and tuned to match an actual experimental cooling appliance. The problem consists in finding an on-off sequence for the adopted optimization horizon and time step that minimizes the energy cost while preserving cooling chamber temperature constraints. To enable the application of metaheuristic optimization algorithms, the problem has to be appropriately encoded. Three different encoding schemes have been designed, suited to both binary and continuous optimization methods. Several metaheuristic algorithms known from the literature are used. Most of them deliver solutions considerably better than a common-sense heuristic compressor schedule. Interestingly, the classical genetic algorithm setup, as well as a setup that was applied to a similar problem in prior research, appear not to work well. The best results are achieved for an alternative genetic algorithm configuration, determined by a series of tuning experiments. Comparable results can be also obtained by the IPOP-CMA-ES or PBIL algorithms, which do not require such extensive tuning and may be preferred by practitioners.**

***Index Terms*—refrigerated warehouse, problem encoding, binary search space**

I. Introduction

Large-scale refrigerated warehouses make it possible to store food for a long time in a safe way. Such storage facilities improve the efficiency, stability, and predictability of production and distribution of various kinds of food. Usually, both the production and consumption of food commodities are seasonal and uneven. The warehouses allow matching the capacity of food processing with the consumers' demand. Examples include production and cooling vegetables during summer for consumption in wintertime or peak consumption of meat during summer barbecues, clearly exceeding production capabilities. Two major kinds of refrigerated warehouses differ with storage temperature – diary or vegetables are usually stored in a mild temperature regime between 1 and $3°C$, while deep-frozen products, e.g., meat or ice-creams, require temperatures strictly below $-18°C$ [1].

A significant drawback of such large-scale refrigerated warehouses is their considerable electricity consumption. Electricity is necessary to drive cooling devices that compensate for all heat losses. Counterbalancing heat fluxes is crucial to control the temperature inside the facilities and keep it within the strict regime regardless of outdoor conditions. Particularly, in summer the differences between indoor and outdoor temperatures might be high, not only in deep-frozen facilities but also in a storage fridge. It causes substantial heat loses even in modern, well-insulated facilities. Moreover, when the outdoor temperature is high or when the outdoor air is moist, the efficiency of standard cooling devices decreases. Consequently the final electricity consumption may be skyrocketing.

Obvious possible solutions, such as installing photovoltaic panels or other additional sources of electricity, are reasonable yet costly. Another answer may be enhancing cooling capacity, using, e.g., ice-bags, phase-change materials, or cryogenics energy storage with liquefied cryogenic gas [2], [3].

In this paper, an alternative approach is adopted, based on shaping the profile of electricity demand by means of short-term storage of thermal energy within the existing infrastructure of the warehouses and within the very outer layer of wrapping or package of deep-frozen products. Sufficient thermal energy storage is provided due to the high value of the available surface and heat capacities of the existing plant's infrastructure, e.g., freezers, cold stores, process water resources, and other elements of the technical infrastructure, through the modification of temperature in time in those facilities. Such a low-cost solution may increase the potential of thermal energy storage, and consequently may enhance power grid sustainability and allow increased production of electricity with renewable sources [2], [3].

The proposed approach requires no investment costs except for the smart energy management system and enhanced measuring equipment. The system optimizes the power demand curve while being fully updated on the actual capability of short-term energy storage in the available infrastructure

Supported by the National Center for Research and Development under grant POIR 01.02.00-00-0355/16.

components, and forecasts the demand for usable energy continuously. Based on such input, decisions can be made on the optimum energy flow in the system for several hours in advance. The system will keep operating at higher efficiency and on a cheaper tariff while relieving the grid during daily peaks. This means a measurable economic benefit for a warehouse owner.

The article formulates the underlying compressor schedule optimization problem and investigates the utility of several metaheuristic algorithms known from the literature. A realistic refrigerated warehouse simulator is used to evaluate the objective functions and verify the constraints. Three different problem formulations are considered: a natural discrete formulation, a direct binarization-based continuous formulation, and an indirect continuous formulation combining target temperature optimization with a rule-based compressor controller. This makes it possible to apply both discrete and continuous metaheuristic search methods. For each configurable algorithm, the best configuration is determined by limited parameter tuning guided by expert knowledge.

A. Related Work

The problem considered in this paper belongs to a broad group of problems related to power and energy systems. According to the surveys [4], [5], the most frequently used optimization methods in that domain are still genetic algorithms. Even though many diverse problems are studied in this application area, the compressor schedule optimization problem addressed by our work has not received much attention in prior research.

In [6]–[8] the authors introduce a refrigerated warehouse controller that is based on an evolutionary algorithm (EA). To the best of our knowledge this is the most closely related prior work to which our research can be naturally compared. The detailed experimental setup is summarized below to make such a comparison easier.

The optimization goal is to minimize the consumption of electrical energy in the horizon of 36 hours. The algorithm optimizes the target warehouse temperature (T_s) for each hour. Therefore, the genotype consists of 36 floats. The implementation provided by the GAlib library [9] is used. The algorithm is configured to use roulette wheel selection, two point crossover, and elitism. The population size is set to 73, the crossover probability is set to 0.6 and the mutation probability is set to 0.05. The algorithm stops after 100 iterations. The optimized vector of target temperatures is converted to a binary sequence of on and off signals for the compressor by a simple controller with hysteresis, i.e., when the temperature in the warehouse exceeds the target value by a specified margin (set to 0.1) then the compressor is turned on. It is turned off when the temperature drops below the target value by more than the same margin.

B. Refrigerated Warehouse Model

A zero-dimensional approach to refrigerated warehouse modeling is adopted. It is based on a straightforward energy balance and heat transfer between the air flowing inside the storage chamber with uniform temperature T_s and the outer layers of store products with uniform temperature T_p.

A sketch of the model of a refrigerated warehouse is presented in Fig. 1 with the following key components: products with cover, indoor air, outdoor air and cooler.

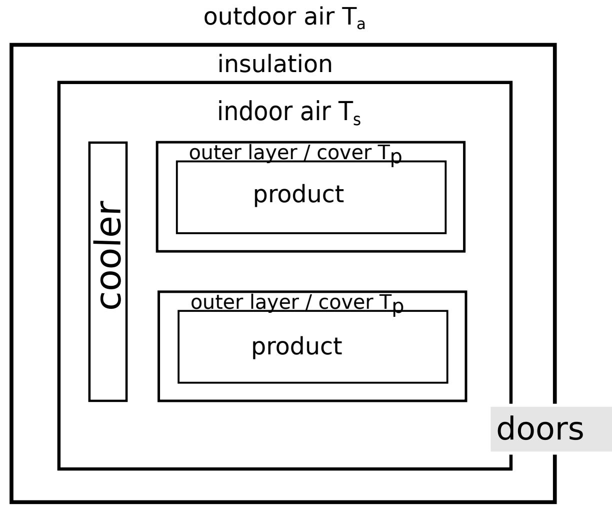

Fig. 1. Sketch of refrigerated warehouse.

Generally all energy losses to the outdoor air with temperature T_a have to be compensated by the internal cooler with the cooling power of $Q_c(\tau)$, usually changing in time (designated by τ) in a complex way. The main assumption is that temperatures T_s and T_p are uniform and depend only on time. This makes it possible to formulate simplified equations of energy conservation in the outer layer of products and in the air inside chamber, T_p and T_s:

$$\frac{dT_p}{d\tau} = \frac{k_p A_p}{m_p c_p}\left[T_s(\tau) - T_p(\tau)\right], \tag{1}$$

$$\begin{aligned}\frac{dT_s}{d\tau} = {} & \frac{k_p A_p}{m_s c_s}\left[T_p(\tau) - T_s(\tau)\right] \\ & + \frac{k_a A_a}{m_s c_s}\left[T_a(\tau) - T_s(\tau)\right] + \frac{Q_c(\tau)}{m_s c_s},\end{aligned} \tag{2}$$

where m_p, c_p, k_p and A_p denote the mass, heat capacity, thermal conductivity, and surface area of the outer layer of products available for heat exchange with the air, m_s, c_s are the mass and heat capacity of the air inside the chamber, T_a is outdoor temperature, and k_a and A_a denote the thermal conductivity and surface area of the wall for heat losses. The fractions occurring in (1) and (2) describe how effectively heat is transferred between the key components of the system.

In the real facilities, the exact values of the effective parameters has to be estimated based on the available historical data or dedicated thermal response analysis. However, the critical challenge is determining the adequate cooling power $Q_c(\tau)$. This quantity combines the effect of several factors, from the power and schedule of compressors to inertia of heat transfer fluid and the details of the coolers themselves. Usually, the first guess of $Q_c(\tau)$ bases on technical documentation. Later, the initial value is adjusted to follow historical data, particularly temperature measurements and compressor schedule.

In the current paper, we adopt an extended numerical model based on [10]. It derives $Q_c(\tau)$ from the given or optimized compressor schedule. It takes into account the whole cooling system: the power and the effectiveness of cold production devices (compressors), the effect of auxiliary tanks filled with phase-change material, and the power and effectiveness of coolers installed in the storage chamber. The Python implementation of the model and its detailed description are available from [11]. The external temperature required by the simulator is taken from the weather forecast.

II. Problem formulation

Compressor schedule optimization can be naturally viewed as a discrete optimization problem, with each candidate solution represented by a binary vector of compressor on or off states. Discrete optimization problems are usually easy to describe and understand but hard to solve. It is practically impossible to determine all possibilities to identify the optimum. Therefore, metaheuristic algorithms that use variants of trial and error methods are frequently used. On the other hand, continuous optimization problems can be easier to solve because some kind of gradient approximation is usually available, which can help to direct the search. Moreover, continuous metaheuristics are developing rapidly, which resulted in several successful parameter-free methods. This is why, besides the natural discrete problem formulation, we also consider two alternative continuous formulations, making it possible to apply continuous search methods with a hope that a better solution can be found without extensive tuning.

A. Discrete formulation

The problem consists in finding an on-off sequence for the next 24 hours that minimizes the energy cost while preserving cooling chamber temperature constraints. According to [8], there is no possibility of continuous control for the compressor, it can be either on or off. The refrigerated warehouse has high inertia and the compressor state cannot be changed too frequently so that discrete-time was assumed with a time step of 15 minutes. Therefore, a cooling schedule can be naturally expressed as a binary vector of length 96, so that discrete optimization methods that operate on binary strings can be used.

B. Direct continuous formulation

One way to present compressor schedule optimization as a continuous problem is to directly transform the discrete problem described above to a continuous representation. According to the survey paper [12], one of the most popular techniques which enable using continuous metaheuristics for binary search spaces is called two-step binarization. In that approach at the first step, the range of continuous variable is reduced to $[0, 1]$ by an S-shaped or V-shaped function. The second step transfers the continuous result of the first step into a binary space, which is called binarization. In the standard binarization approach, the outcome of the transfer function is treated as the probability of the *truth* value. A binary string is created by sampling from the $\{0, 1\}$ set according to such probabilities. In this paper we skip the transfer function step, because the optimization algorithm will be working on continuous variables limited to the $[0, 1]$ range.

Modifying *truth* value probabilities and changing them to a binary string by sampling is also the principle of the PBIL algorithm [13]. PBIL uses simple heuristics to modify the probabilities. From that point of view, the proposed approach uses a continuous search method to replace PBIL's heuristics. PBIL will be described in more detail in Section III-A2.

C. Indirect continuous formulation

Another possibility to use continuous search methods for compressor schedule optimization is to adopt an indirect approach in which a sequence of warehouse target temperatures is optimized and then used to generate on and off signals for the compressor. More specifically, each candidate solution is represented by a sequence of target warehouse temperatures (T_s) for each time interval of the 24-hour optimization horizon. Decisions to turn the compressor on or off are made by a simple deterministic controller based on these temperatures. Two lengths of the interval will be considered: one hour and fifteen minutes. One hour intervals match assumption of the related work discussed in Section I-A. The interval of fifteen minutes is used to match the previously presented discrete formulation of the problem. Considering two time intervals makes it also possible to examine how continuous methods scale with the increase of the problem dimensionality.

In this study the controller operates in discrete time and can change the state of the compressor once for a quarter. That behavior is physically justified and it does not require specifying an additional temperature margin, which is usually required to implement hysteresis. This is unlike in the related work [6]–[8] with a deterministic controller working in continuous time and therefore using hysteresis.

Our controller operates using simple decision rules at each time step. The compressor is turned on when the temperature in the warehouse exceeds the target value or the upper constraint and turned off when the temperature drops below the target value or the lower constraint. Notice that, due to the discretization of time and large inertia of the cooling hardware and the warehouse, this does not guarantee the feasibility of the solutions. Therefore, temperature constraints must be still handled by the applied optimization methods, just like with the other problem formulations.

III. Metaheuristic algorithms

This paper examines the suitability of several different metaheuristic optimization methods to the compressor schedule optimization problem, assuming both binary and continuous multidimensional search spaces. They are briefly reviewed below to provide the necessary background information, literature references, and specify variants of particular algorithms used for this work.

A. Metaheuristics for binary search spaces

In the discrete problem formulation the compressor schedule is represented by a bit string. Therefore, metaheuristics that operate on binary strings match the problem perfectly.

1) Genetic Algorithm: Genetic algorithms (GAs) are quite old methods [14], [15] but they may be still useful for discrete search spaces. In general, at each iteration of a GA, the selection, mutation, crossover, and succession operations are performed. There are many ideas in the literature about how to implement each of those operations. In the classical GA a roulette wheel selection, one-point crossover, and generative succession are used. Apart from the specification of components, the GA has several parameters to set up, i.e., the population size, the probability of performing mutation, and the probability of crossover. If the latter is 0 then only mutation is used and if the latter is 1 then all candidate solutions are the result of crossover.

Apart from the classic GA, configurations with different components were experimentally verified. Because of the page limit, only the results of the best configuration are presented in the paper. The best result was yielded by a version with exponential rank selection (ERS), uniform crossover, and elitist succession. In ERS [16] individuals are ranked according to fitness and then the probability of selection of the individual is decreased exponentially in the function of the rank. In uniform crossover a donor of each bit of the child is randomly selected from the parents. The elitist succession ensures the survival of e best solutions from the sum of the current and candidate population. For both the classic GA setup and the alternative (ERS, uniform crossover, and elitist succession) setup mutation is performed by bit negation.

2) PBIL: Probability-Based Incremental Learning (PBIL) [13] maintains a real-valued vector that represents the probability of each bit being one. At each iteration, the algorithm generates random solutions according to the current probability vector. In the classical PBIL version from the set of generated solutions only the best one is used to update the probability vector, so as to increase the chances of generating that solution. The algorithm has two parameters: the sample size, which determines how many samples are generated at each iteration, and the learning rate, which determines the weight of the current update of the probability vector.

There are variants of PBIL that can improve search effectiveness. These variants introduce mutation of each component of the probability vector (to inhibit premature convergence) and learning from the worst solutions (update probabilities in a way to avoid them).

B. Metaheuristics for continuous search spaces

Apart from methods specially designed to operate on bit-strings, methods designed for continuous optimization will be used here as well, with the previously presented direct and indirect continuous problem formulations. The recent development of continuous metaheuristics is directed towards parameter-free methods with auto-adaptation. Therefore, their application is straightforward, even for practitioners without experience in the application of metaheuristics.

1) IPOP-CMA-ES: The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [17] is an efficient derivative-free optimization algorithm. At each iteration, the neighborhood of the current working point is sampled using a multivariate normal distribution specified by the covariance matrix. The mutation strength is controlled by a parameter called step size. At each iteration the current working point, the covariance matrix, and the step size are updated to maximize the probability of successful moves.

Since it is a very robust but still local optimizer, it could be trapped in local optima. Therefore, to increase the chance of finding a global optimum, a restart is performed whenever stagnation is detected. In the IPOP-CMA-ES [18] variant at each restart the population size is automatically increased. In the remainder of the paper the name of this method will be shortened to IPOP. The implementation of IPOP is available in the `cma` Python package.

2) Differential evolution: Since the first publications [19], [20] presenting the algorithm, differential evolution (DE) has received increasing attention, which can be attributed to its simplicity and efficiency. In this paper its version named DE/local-to-best/1/bin is used. In that version the i-th mutant is a result of the sum of i-th solution, the difference between two randomly selected solutions, and the difference of the best solution in the current population and the i-th solution, i.e.: $v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r1} - x_{r2})$, where the scale factor F (usually $F \in (0, 1]$) is a parameter of the algorithm. The candidate solution is the result of crossover if a random number sampled from the standard uniform distribution is less than CR and it is equal to the mutant otherwise. The CR is a parameter of the algorithm. In this paper, a binomial crossover is used (bin). It creates an offspring by discrete recombination of the mutant vector v_i and the parent vector x_i.

3) jSO: The jSO algorithm [21] is a version of DE which does not require parameter tuning and component selection. It uses the DE/current-to-pBest-w/1 mutation strategy $v_i = x_i + F_w \cdot (x_{pbest} - x_i) + F \cdot (x_{r1} - x_{r2})$, where F_w is $0.7F$ for the first 20% of fitness evaluations budget, then it is $0.8F$ until 40% of the budget, and then it becomes $1.2F$. The x_{pbest} is randomly selected from the set of the A best solutions in the population, where A is a parameter that is also adapted during evolution. The F value is randomly drawn from the standard Cauchy distribution and CR is drawn from the standard normal distribution. The location parameters of both distributions are also adapted during the search.

IV. Experimental study

The experimental study is composed of three parts. The first part compares results of two binary optimization methods with a result of a common-sense solution. In the second part two modern continuous optimization metaheuristics are used with the direct continuous problem formulation, based on the standard binarization method. The results are compared with the results of PBIL. In the third part several continuous optimization methods are applied to the indirect continuous problem formulation, i.e., to optimize the target temperature inside the warehouse (T_s) rather than the on-off schedule. The optimized target temperature sequence is converted to a bit string representing a compressor schedule by the simple deterministic controller described in Section II-B. The results are compared with the results of the best method optimizing the compressor schedule directly.

The fitness function to be minimized is the weighted sum of energy cost c of the compressor schedule and quadratic penalty p for temperature constraints violation:

$$q(x) = c(x) + \alpha p(x), \tag{3}$$

$$p(x) = \sum_{j:u<t_j} (t_j - u)^2 + \sum_{j:l>t_j} (l - t_j)^2, \tag{4}$$

where $\alpha = 10^9$, j is a time step index, t is the temperature inside the warehouse, l is the lower temperature bound equal to 1°C and u is the upper temperature bound equal to 3°C. In this study, a two-state energy tariff is adopted because it is used in reality and a common-sense heuristics can be proposed for it. Fig. 3 a) presents the energy tariff and the external temperature used for the experiments. A Python implementation of the fitness function and refrigerated warehouse model is available at [11].

The experimental methodology used in the paper was inspired by the rules of the CEC benchmark for single objective bound constrained problems [22]. All methods were working on the same fixed budget which was set to 10000 objective function evaluations. The budget cannot be easily increased because of the expensive objective function calculation.

All methods start their operation from a randomly initialized starting point (IPOP) or population (EA, GA, DE, jSO). The results of 25 independent runs of each optimization method were collected and presented in the form of boxplots.

A. Common-sense schedule

Having a common-sense schedule provides a valuable reference baseline for the results of the optimization. According to common sense and some expert knowledge about refrigerator systems, the inside temperature should be generally held near the upper limit to minimize energy loss through the walls and doors and to achieve better cooling effectiveness. On the other hand, using energy during the high price period should be strongly avoided as the high price is greater than the low price by a factor of more than two. Therefore, some time before the high price period the compressor should be turned on to cool the air to the lower bound. As a result, the compressor could be turned off during the high price period until the temperature approaches the upper bound. The compressor schedule that arises from the aforementioned schema is presented in Fig. 3 b), along with the resulting temperature inside the warehouse.

B. Binary search space methods

Besides the classic GA, several GA configurations with different components were experimentally verified. The experiments included examining selection methods: roulette wheel, tournament of size 2, rank selection, exponential rank selection; succession methods: generative, elite, adopted from Differential Evolution; crossover methods: one point, two points, uniform; expected number of mutated bits: 1, 2, 3, 4; crossover probability: 0, 0.5, 0,8, 0.95, 1; population size: 10, 20, 40, 80, 160, 200, 320, 640, 1280. Because of the page limit, only the results of classical GA and the best alternative configuration, using exponential rank selection with a base of the exponent equal to 0.75 (ERS), uniform crossover (UNI), and elitist succession of size 1 (ELI), are presented. For both versions, the population size was set to 200, the probability of mutation to 0.01, and the probability of crossover to 0.95.

Apart from the classical PBIL, results of several alternative versions were also examined, which included the following settings: the number of the best solutions used: 0, 1, 2; number of the worst solutions: 0, 1, 2, 3; sample size: 5, 10, 20, 50, 100; learning rate: 0.05, 0.1, 0.2; mutation probability: 0, 0.05, 0.1, 0.15, mutation range: ± 0.01, ± 0.02, ± 0.03. The best identified configuration uses the two best and the two worst solutions, sample size 20, learning rate 0.05, mutation probability 0.1, and mutation range ± 0.02.

The results of the aforementioned algorithms along with the results of the common-sense solution (marked by dotted line) are shown in Fig 2.

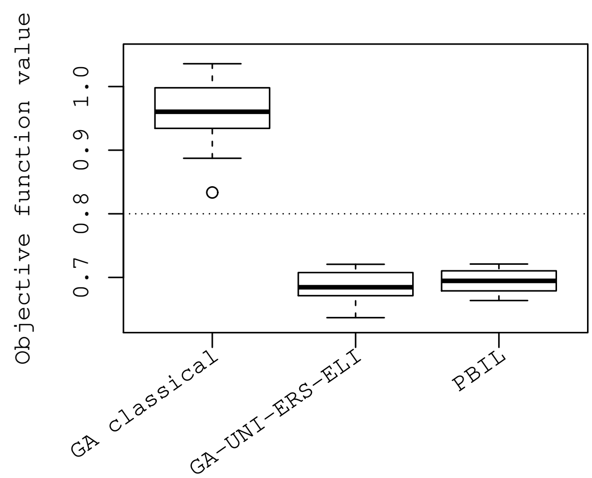

Fig. 2. Comparison of methods that operate on a bit-string. The result of the common-sense solution is marked by a dotted line.

It can be seen that the classical GA gives worse results than the common-sense solution. On the other hand, the GA version with uniform crossover, exponential rank selection and elitist succession (GA-UNI-ERS-ELI) gives the best result, which is substantially better than the common-sense solution. The results of PBIL are comparable to the results of GA-UNI-ERS-ELI.

The best solution optimized by GA-UNI-ERS-ELI, along with the inside warehouse temperature that stems from its application, is shown in Fig. 3 c). To facilitate the comparison, the commons-sense solution and the result of its application is presented in Fig. 3 b). The energy price and the temperature outside the warehouse are shown in Fig. 3 a). As it can be observed, using expensive energy should not be avoided so strongly as the common-sense schedule assumes. It is more efficient to cool down when the difference between the inside and outside temperatures is small. It is also more efficient to turn on the compressor periodically than to turn it on for a long period of time, which results in the air temperature reaching the lower bound.

C. Continuous methods with standard binarization

This section presents the results of two modern continuous optimization metaheuristics (IPOP and jSO) coupled with standard binarization method. These metaheuristics usually do not require tuning because they adapt most of their parameters and provide reasonable values for the rest of the parameters. The results are compared with the results of PBIL, because it uses the same binarization method.

The results PBIL and IPOP are presented in Fig 4.

It can be seen that the results of IPOP are much worse than those of PBIL. The results of jSO were not shown on the plot because the algorithm was not able to find a feasible solution what is the cause of achieving a poor average fitness value 3.29E+08, which is however still better than 1E+09 – the average result of Monte Carlo search.

The reason for the poor performance of IPOP and failure of jSO is that stochastic conversion of a probability vector into a bit string makes the objective function very noisy, i.e., the same continuous solution results in very different bit strings on subsequent evaluations, often leading to substantially different fitness values.

D. Continuous methods optimizing the target temperature

This section presents the results of the optimization of target warehouse temperatures (T_s) for each time interval by the following continuous metaheuristics: IPOP, DE, jSO.

1) One hour intervals: As mentioned in Section I-A a similar problem was solved in the Night Wind (NW) project [6]–[8] using a standard evolutionary algorithm to optimize T_s for each hour. For the comparison purpose in this experiment we will also use one hour intervals. This gives a 24-dimensional continuous search space as we are interested in optimization for a day ahead. The results of IPOP, DE and jSO were compared to the results of the evolutionary algorithm configured as in the NW project (EA-NW) and to the result of the best discrete method. In EA-NW the population size was set to $2n + 1$, where $n = 24$ is the problem dimensionality. The crossover probability was set to 0.6 and the mutation probability was set to 0.05. Besides that, the version with population size $\mu = 20$ was also examined (EA-NW-20) as that population size yields the best results for the DE algorithm.

For this problem, several combinations of DE parameters were experimentally verified, including CR: 0.1, 0.5, 0.9; F: 0.8, 0.9; μ: 10, 20, 40. As for GA, only the results of the best configuration, which is F=0.9, CR=0.5, μ=20, is shown here. The parameters of IPOP and jSO were left default. The results of the experiments are shown in Fig 5.

As it turns out, none of the continuous methods is better than the best method that operates on bits (GA-UNI-ERS-ELI). Rather surprisingly, from the set of evaluated continuous methods, EA-NW is the worst. It gave much better results after reducing the number of individuals to 20 (EA-NW-20), but it still remains the worst among considered methods.

Both the modern continuous methods, i.e., jSO and IPOP, gave similar results, with a slight advantage of IPOP. What is interesting here, the simple and much older DE is the best continuous method in the experiment. The comparison of the results of DE and jSO leads to the conclusion that sometimes it may be better to configure the DE based on human experience and limited tuning experiments rather than leave configuration to an automatic process like in the case of jSO.

2) Fifteen minutes intervals: The comparison in which the GA operates in a 15 minute time resolution but continuous methods operate in an hour resolution can be considered unfair, because continuous methods have 4 times fewer parameters to optimize. On the other hand, the controller used for continuous methods to convert optimized target temperature sequences to on-off signals could introduce some non-optimal behavior. To dispel these concerns, the modern continuous methods were run again with a 4 times longer genome, i.e., they optimized T_s for every 15 minutes. The results of this experiment are presented in Fig. 6.

It turns out that jSO scaled poorly but IPOP gave even slightly better results than previously for the shorter genome. Presumably, the 15 minute time resolution permitted it to find a better control strategy than that obtained by conversion from hourly target temperature values by the simple deterministic controller.

V. Conclusions

In this paper, the problem of cooling schedule optimization was considered. The problem was solved using both binary and continuous metaheuristic optimization algorithms. The former directly optimized the compressor on-off schedule using a natural discrete problem formulation, whereas the latter were used with continuous problem formulations in 3 scenarios: 1) to optimize probabilities of turning the compressor on; 2) to optimize T_s for each hour; 3) to optimize T_s for each quarter.

The results of the experiments make it possible to draw several interesting and useful conclusions. The most important is that the optimization result is much better than the common-sense solution even for a simple two-state electricity price tariff. Even greater advantage of the optimized solutions can be expected with more complex tariffs or dynamic market prices. This confirms the high practical utility of compressor schedule optimization for refrigerated warehouses and encourages further work in that direction.

The best results are achieved by a genetic algorithm with components and configuration selected using human expe-

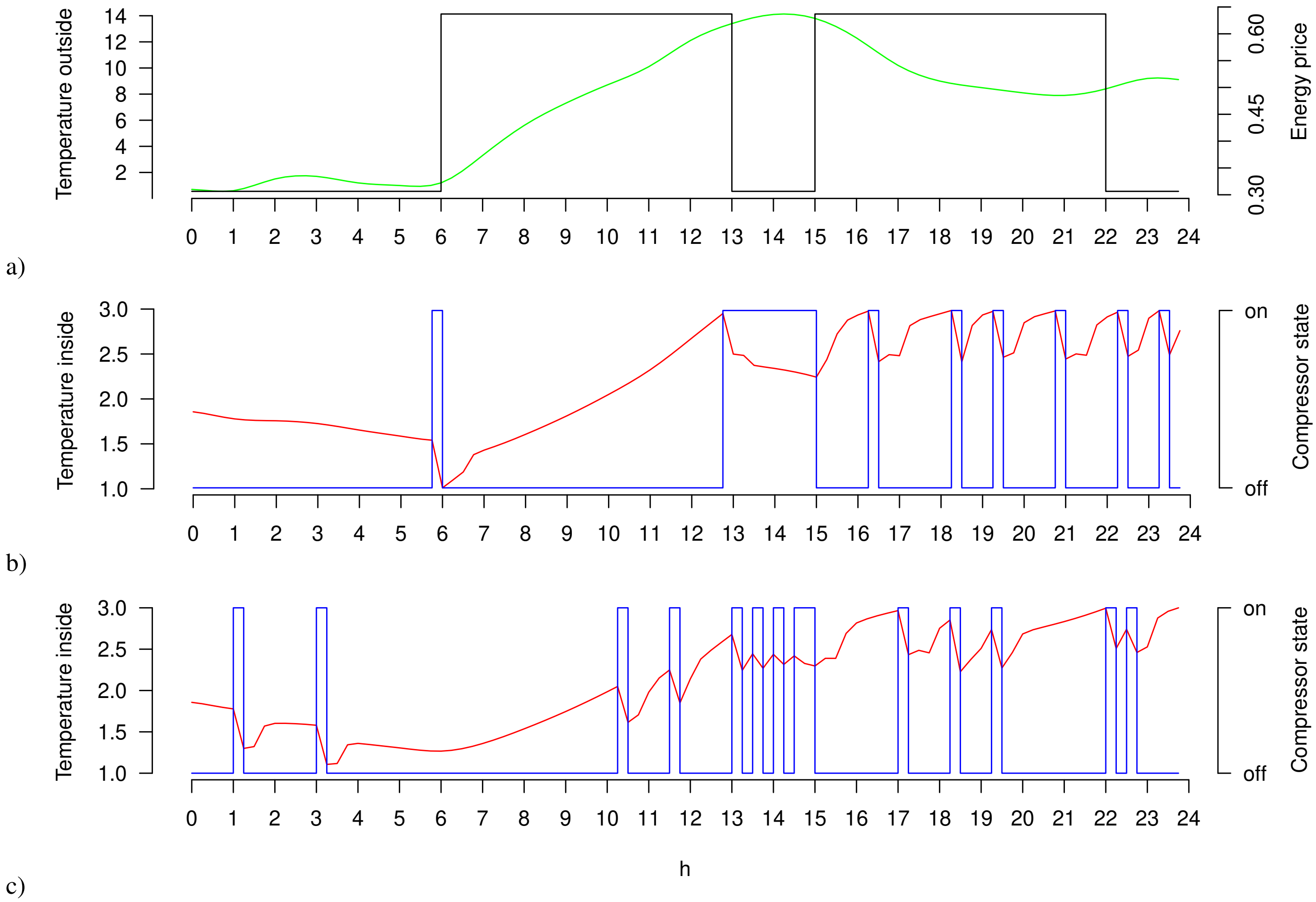

Fig. 3. The air temperature outside the warehouse and the energy price (a); the common-sense compressor control schedule and temperature inside the warehouse (b); the GA-UNI-ERS-ELI optimized compressor control schedule and the temperature inside the warehouse (c).

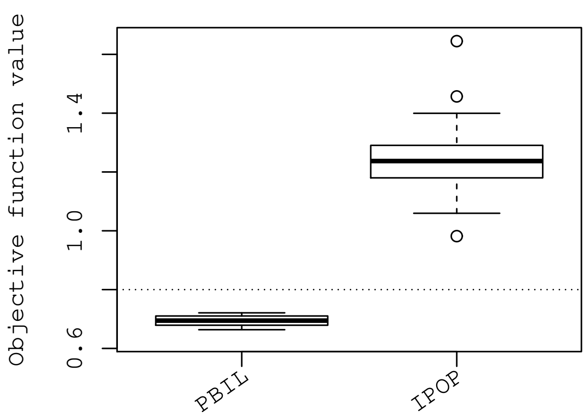

Fig. 4. Comparison of selected metaheuristics that work on the probabilities of *true* vale. The result of the common-sense solution is marked by a dotted line.

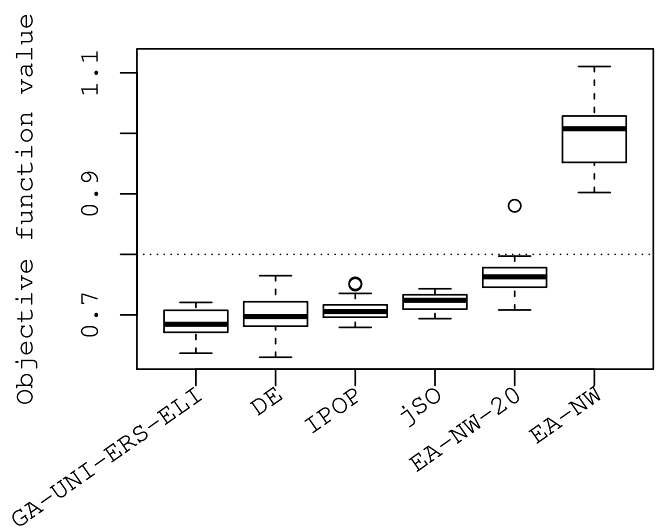

Fig. 5. Comparison of selected continuous metaheuristics optimizing the target temperature for each hour and the best algorithm from the binary formulation of the problem (GA-UNI-ERS-ELI). The result of the common-sense solution is marked by a dotted line.

rience and confirmed by limited experiments. The obtained results are much better than those of the classical GA. Even though the GA is a very old concept, it is interesting to see

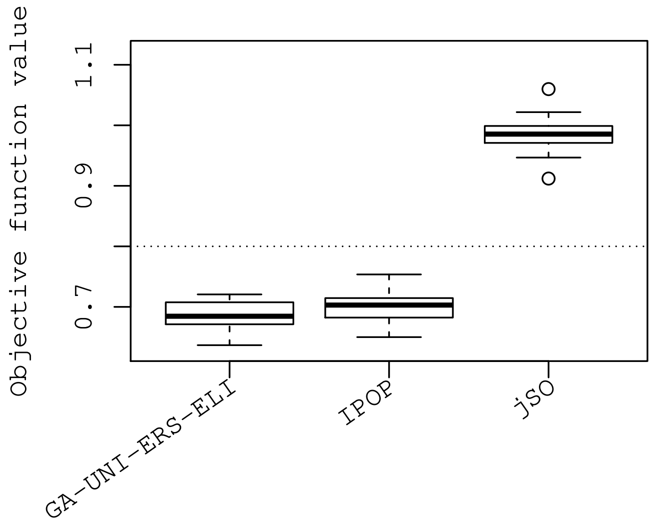

Fig. 6. Comparison of selected metaheuristics optimizing the target temperature with a 15-minute resolution. The result of the common-sense solution is marked by a dotted line.

that some of its variants can still be successfully applied to challenging real-world problems.

The other algorithm that deserves special interest and appreciation is PBIL. Even though its implementation and application are much simpler than those of GA, it reached a similar level of solution quality.

Using the standard binarization method with modern metaheuristics turns out a wrong idea because it makes the objective function very noisy and, consequently, the problem hard to solve.

Using off-the-shelf contemporary continuous optimization methods to optimize T_s gave slightly worse results than the GA. It is worthwhile to notice, though, that IPOP required no configuration and its implementation is easily available. Therefore, it may be still a good choice for practitioners that do not want to experiment with different GA setups.

References

[1] G. F. Hundy, *Refrigeration, air conditioning and heat pumps*. Butterworth-Heinemann, 2016.

[2] S. Rosiek, M. S. Romero-Cano, A. M. Puertas, and F. J. Batlles, “Industrial food chamber cooling and power system integrated with renewable energy as an example of power grid sustainability improvement,” *Renewable Energy*, vol. 138, pp. 697–708, 2019.

[3] K. Fikiin, P. Mazzucchelli, A. Foster, J. Evans, S. Akterian, G. Alvarez, D. Leducq, M. Hoang, G. Garcia, L. Varga *et al.*, “Energy mapping of large refrigerated warehouses co-located with renewable energy sources across europe,” in *ICR 2019: The 25th IIR International Conference of Refrigeration*, 2019.

[4] K. Y. Lee and Z. A. Vale, Eds., *Applications of Modern Heuristic Optimization Methods in Power and Energy Systems*. Wiley-IEEE Press, 2020.

[5] J. Soares, T. Pinto, F. Lezama, H. Morais, and P. De Lellis, “Survey on complex optimization and simulation for the new power systems paradigm,” *Complexity*, vol. 2018, Jan. 2018.

[6] H. W. Bindner, T. Cronin, and Y. Zong, “Night Wind deliverable D.3.2. Main simulation report,” Danmarks Tekniske Universitet, Tech. Rep., 2008.

[7] Y. Zong, T. Cronin, O. Gehrke, H. Bindner, J. C. Hansen, M. I. Latour, and O. U. Arcauz, “Application genetic algorithms for load management in refrigerated warehouses with wind power penetration,” in *2009 IEEE Bucharest PowerTech*, June 2009, pp. 1–6.

[8] Y. Zong, H. W. Bindner, and O. Gehrke, “Different optimal control strategies for exploitation of demand response in the smart grid,” *IFAC Proceedings Volumes*, vol. 45, no. 25, pp. 188 – 193, 2012, 15th IFAC Workshop on Control Applications of Optimization.

[9] *GAlib – a C++ library of Genetic Algorithm components*, Jan. 2021. [Online]. Available: http://lancet.mit.edu/ga/

[10] R. Kwidziński, “In Polish: Model dynamiki temperatur w komorze przechowalniczej z materiałem zmiennofazowym w pośrednim obiegu chłodniczym,” Institute of Fluid-Flow Machinery Polish Academy of Sciences, Tech. Rep., 2015.

[11] *Implementation of the model of refrigerated warehouse and objective function*, Jan. 2021. [Online]. Available: https://staff.elka.pw.edu.pl/%7Erbiedrzy/publ/rwm.zip

[12] B. Crawford, R. Soto, G. Astorga, J. Garcia Conejeros, C. Castro, and F. Paredes, “Putting continuous metaheuristics to work in binary search spaces,” *Complexity*, vol. 2017, pp. 1–19, 01 2017.

[13] S. Baluja and R. Caruana, “Removing the genetics from the standard genetic algorithm,” Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-95-141, May 1995.

[14] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989.

[15] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI, USA: University of Michigan Press, 1975.

[16] P. J. B. Hancock, “An empirical comparison of selection methods in evolutionary algorithms,” in *Evolutionary Computing*, T. C. Fogarty, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 80–94.

[17] N. Hansen and A. Ostermeier, “Completely derandomized self-adaptation in evolution strategies,” *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[18] A. Auger and N. Hansen, “A restart CMA evolution strategy with increasing population size,” in *2005 IEEE Congress on Evolutionary Computation*, vol. 2, Sept 2005, pp. 1769–1776 Vol. 2.

[19] R. Storn and K. Price, “Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces,” TR-95-012, ICSI, Tech. Rep., 1995.

[20] ——, “Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, Dec 1997.

[21] J. Brest, M. S. Maučec, and B. Bošković, “Single objective real-parameter optimization: Algorithm jSO,” in *IEEE Congr. Evol. Comput.*, 2017, pp. 1311–1318.

[22] P. N. Suganthan *et al.*, “Problem definitions and evaluation criteria for the CEC 2005 Special Session on Real Parameter Optimization,” Nanyang Technological University, Singapore, and KanGAL, IIT Kanpur, India, Tech. Rep., 2005.