Intelligent Scheduling of the Heat Pump

by

Behzad Lashkari

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Software Engineering and Intelligent Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

Space heating is responsible for a significant portion of energy consumption in the residential sector. As such, it has a great potential for energy savings. The heat pump is a heating device that offers important energy conservation properties. It allows modification of residential energy demand profile and subsequent reduction of electricity consumption and costs. Living comfort of the residents is the other side of the coin that also must be considered in the heat pump scheduling optimization process. This thesis presents an intelligent approach to heat pump scheduling problem based on metaheuristic optimization algorithms. In particular, we consider mutation-based binary particle swarm optimization (M-BPSO) and genetic algorithm (GA). Since standard BPSO suffers trapping in local minima, it has been augmented with a mutation operator. However, mutation alone can not effectively address all the shortcomings of standard BPSO. Therefore, other enhancements of the algorithm have been implemented to improve its overall performance. Performance of all considered algorithms is evaluated using a series of simulation experiments and thoroughly compared. The simulation results confirm that the proposed approach can optimize the heat pump schedule without sacrificing the thermal comfort of residents, and that the improved algorithm can obtain the optimal schedule with high efficiency.

I would like to dedicate this thesis work to my parents.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Petr Musilek, for his invaluable comments, great patience, and support during my study, and his dedicated work editing my thesis.

Also, I would like to extend my special thanks to Dr. Yuxiang Chen for all of his support throughout graduate school. I am also thankful to all committee members for taking the time to review my thesis, and for their valuable comments and suggestions.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In order to slow down global warming phenomena governments in many countries have set up decarbonisation goals to supersede fossil fuel with “green” energy sources. To this end, electricity plays an important role and the energy demand characteristics are expected to change in the near future. In particular, the electrification of the transportation sector (through dispersion of hybrid and electric vehicles) and electrification of the heating systems of buildings (using heat pumps and/or electric heaters) will likely bring significant changes to the residential energy consumption profiles [1]. According to the prediction by National Grid, UK households may deploy 9 millions electric heat pumps by 2030. This would shift a portion of heating energy consumption from gas to electricity [2]. To achieve sustainable, efficient and reliable utilization of the electric system assets, this changed reality will require new technologies and applications.

End-users are also expected to play a new, more active role in the future energy scenario. Many will turn into so-called prosumers, who both consume and produce electricity. However, due to the uncertainty and volatile nature of renewable energy sources, the mismatch of renewable supply and demand can be expected. This may considerably alter the electricity production/consumption principle, from “generation following demand” to “demand following generation” [3]. On this subject, in addition to installation of renewable energy sources [4], many consumers will also add more flexibility and facilitate efficient energy management through modifications on the demand side of the energy balance equation. Such flexibility can be achieved, for example, by postponing or scheduling running time of certain residential loads. Demand side management (DSM) is modification of consumer energy demand to support flexibility of the energy system [5]. The level of flexibility is evaluated by the amount of controllable power and the time for which the operation of the end-user loads can be delayed [6]. In general, DSM programs can be classified in two main categories [7]:

- price-based programs, where customers are motivated to adjust their demand pattern in response to the day ahead or real-time price signals.
- incentive-based programs, where customers allow utilities to directly control or schedule some of their loads and are rewarded through specific incentives in the tariff scheme.

Among various appliances in the residential sector, electro-thermal devices installed for space heating are suitable candidates for use in DSM schemes [8]. This is mainly due to the thermal energy storage capabilities of the buildings, which leads to relatively slow thermal dynamics and interruption possibilities of the electro-thermal device with slight effect on comfort of the residents. Due to the electrification of heating systems and residential energy efficiency targets, the design of DSM methods for controlling such heating systems has recently attracted much researcher interest [6]. This study presents an intelligent approach for the day-ahead scheduling of heat pump using genetic algorithm (GA) and mutation based binary particle swarm optimization (M-BPSO).

This document is organized as follows. Chapter 2 reviews heat pump scheduling approaches proposed in the literature and briefly points out the advantages and disadvantages of propounded solutions. Chapter 3 outlines the essential theoretical background regarding metaheuristic optimization and the heat pump technology.

Chapter 4 presents the model parameters, configuration of the exerted algorithms, objective function and preserved constraints. Chapter 5 demonstrates the simulation results in different scenarios together with discussion and analysis. Chapter 6 sums up the achievements of this investigation and provides potential future directions.

Chapter 2

Related Work

The idea of using electro-thermal devices as DSM instruments has recently attracted much researcher interests. The motivation behind the advent of DSM methods for managing the heat pumps has been triggered by two main attributes. First, due to energy efficiency enhancement, electric heat pumps are anticipated to be deployed massively for space heating in the near future [9]. At the beginning of 2011, the number of installed heat pumps supplying heat to Germans' buildings was estimated to be more than 350,000 [10]. As they spread on a large scale, the electricity demand is expected to alter. J. Love et al. [11] indicated that 20% growth in households using heat pumps leads to a 14% increment in peak grid demand. Such changes need to be managed to maintain sustainable and reliable grid. Second, slow thermal variation due to thermal storage capabilities of the buildings leads to the flexible running time of heat pumps, which can be exploited towards smart controlling of such electro-thermal devices. This flexibility can be utilized by price-based DSM methods for electricity cost minimization and for mitigating issues that may arise in the distribution grid.

There are many studies reported in the literature, that pay particular attention to various aspects of designing DSM methods for controlling heat pumps. The first challenge for projecting such approaches is determining proper thermal models to demonstrate the thermal dynamics of buildings and heating systems. A thermal model of under floor heating system coupled with air source heat pump was developed in MATLAB/SIMULINK [12]. A general scheme for modeling the electro-thermal domestic heating system and building based on an electrical analogue is represented in [13, 14]. In some studies, heat pumps are coupled with thermal energy storage (TES) systems to attain further flexibility in proposed DSM schemes. The performance of heat pump coupled with TES system in the form of stratified hot water tank was investigated by A. Arteconi et al. [15]. Simulations were performed in the TRNSYS [16] environment and the results showed an extension in the heat pump's off time, up to 3 hours.

Various DSM methods can be designed based on different objective considerations, such as minimizing the cost of electricity for the end-user or maximizing the consumption of self-generated power in presence of renewable supply. A DSM method for controlling the heat pump to adapt to wind energy generation on the household level has been introduced by M. Diekerhof et al. [17]. In other work, M. Diekerhof et al. [18] considered solar energy generation in their proposed approach for smart scheduling of electro-thermal heating units, and introduced an algorithm to maximize the renewable energy utilization while minimizing CO_2 emissions.

As previously mentioned, many of the proposed DSM methods in the literature aim to minimize the energy cost for final customer based on price-based model. While such methods are usually considered as a service to the end-users, utilities can also benefit from them and manage the grid by transmitting various price signals over time and count on the feedback of price-responsive consumers. C. Molitor et al. [19] investigated different operating modes of heat pumps: the heat-driven and electricity-price-driven modes. In heat-driven operating mode, the heat pump runs whenever the thermal demand is not satisfied, while the electricity-price-driven mode attempts to shift the heat pump running time to low price intervals. Authors have also considered two different tariff schemes. Accordingly, a heat pump tariff offered by German energy provider and a time-varying electricity tariff were considered as underlying schemes. Simulation results indicated that optimum scheduling of heat pump based on electricity-price-driven mode led to significant electricity cost reduction. However, this was achieved at the cost of high fluctuation of indoor temperature and comfort loss. M. Loesch et al. [20] proposed an evolutionary based algorithm for scheduling the heat pumps and electricity cost minimization. The main idea behind the introduced approach is overheating the hot water tank to enhance the degree of freedom (DOF) in scheduling procedure. The authors performed simulations by considering the time-variable electricity price scheme and maximum power consumption constraint. The presented algorithm exploits the flexibility in heat pump's running time towards minimizing the customer's electricity cost. However, the link to the thermal comfort delivered to occupants has not been taken into account. In another investigation [21], a two-level control process is considered to realize the flexibility offered by heat pumps using interconnected 300 liters and 500 liters storage tanks. In the first level, heat pumps are scheduled with respect to the time-varying electricity price signal targeting cost reduction. In the second level, the real-time controller adjusts the predefined operating state of heat pumps based on the network conditions (voltage limit). The authors evaluated the performance of their proposed algorithm with respect to cost reduction and load adaption boundaries. While some comfort loss is inevitable due to heat pump's operation shifting, no details about the occupant comfort are provided in this work. L. Zhang et al. [22] introduced an algorithm to adjust the electric heat pump's operation time with respect to renewable generation profile, aiming at maximizing the usage of renewable energy sources. The proposed "generation match algorithm" is evaluated considering the comfort level of occupants. Results indicated that the proposed method drastically affects the comfort of occupants in both wind and solar photovoltaic (PV) generation scenarios. However, the solar PV scenario has suffered a relatively higher level of discomfort in comparison with the other scenario.

Several studies addressed the thermal discomfort of occupants caused by reallocating the heat pump operation time. The main idea behind these approaches is enforcing the constraints on the thermal comfort delivered to occupants during the heat pump scheduling process. F. D. Angelis et al. [23] proposed a method based on mixed-integer linear programming paradigm for task and energy scheduling. In their study, the optimization goal is minimizing the cost of electricity by optimal scheduling of flexible household appliances, including heat pump. The authors consider the occupant's comfort in terms of indoor temperature boundaries. They also introduce a constraint on the maximum amount of electricity that can be purchased from the grid, to maximize the utilization of renewable energy sources. In addition, They consider one hour time interval resolution in their simulation. Results show that the proposed algorithm violates indoor temperature limit in several cases. Moreover, the authors used a simplified thermal model of the heat pump, negatively affecting the accuracy of simulations. In other studies [4, 24], authors presented an optimization algorithm aiming at power peak shaving and fulfilling the customer's thermal comfort preferences. Results indicated that the proposed algorithm maintains the indoor temperature within predefined range while reducing the power peaks. It is worth noting that, authors introduced an approximation in heat pump modeling which can lead to inaccuracies in simulation results.

To the best of our knowledge, the thermal discomfort of occupants due to the heat pump scheduling for electricity cost minimization has not been effectively addressed by previous studies. This was the main motivation to develop the intelligent approach for heat pump scheduling described in this thesis. The presented scheduling method only uses the thermal storage capacity of the building, without any additional thermal energy storage systems. The proposed approach is evaluated under several scenarios, while considering different restrictions on the indoor temperature variation and using realistic electricity pricing. The results show that the proposed method can take advantage of the heat pump flexibility towards to reduce electricity costs while maintaining the thermal comfort of the occupants.

Chapter 3

Background

3.1 Heat Pump

The notion behind the heat pump is the transmission of thermal energy from a heat source to a heat sink in which heat flows from a low-temperature environment to a high-temperature environment [25]. The thermal energy flow of the heat pump is in the inversed direction of the spontaneous heat flow and requires energy expenditure. The heat pump operates by extracting the accessible heat from the low temperature source side and augmenting its temperature in a manner to be useful for space heating. While the concept of water pump which transfers the water from low pressure (downhill) location to high pressure (uphill) location is clear for us and represents a similar concept as heat pump, understanding the heat pump procedure seems to be more challenging. One of the common approaches for facilitating the conceptual difficulties associated with the heat pump is making reference to a familiar home appliance. Refrigerators also extract heat from the low-temperature environment (food compartment) and transfer it to the high-temperature environment (kitchen). The radiators on the back of the refrigerator, transmit the heat from the fridge's interior to the exterior environment using hundreds of watts of electricity that is required for powering the compressor. Most heat pumps, including refrigerators, transfer heat by circulating a refrigerant fluid through a compression-expansion cycle (Fig. 3.1).

The refrigerant cycle, illustrated in Fig. 3.1, includes four steps as follows [25]

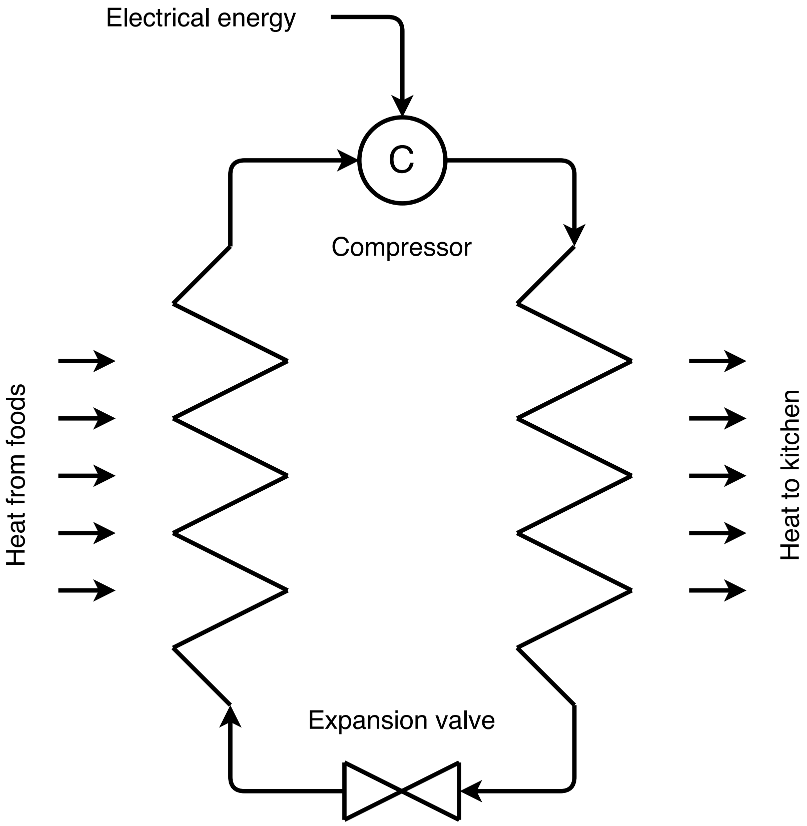


Figure 3.1: Refrigerant cycle

1. The refrigerant liquid is moving around the low temperature cabin of fridge (environment) through a network of pipes known as evaporator. The refrigerant is chosen in a manner that will boil at a temperature below 0°C under the pressure conditions of cycle. The boiling refrigerant then evaporates by absorbing the required vaporization heat from the fridge's interior.

2. The refrigerant, which is now a warmer vapour, will be pressurized and reaches to higher temperature level by passing through a compressor.

3. In next step, refrigerant passes through another heat exchanger called condenser, which is the radiator grid on the fridge's back side. By passing through condenser, the vapour transfers its heat to the kitchen and will change to liquid due to heat losses.

4. The cycle will be completed when compressed refrigerant passes the expansion

valve and, as the result, its pressure and temperature drops significantly, back to the initial value.

A heat pump, employed for space heating, operates through the same cycle; only the value of pressures and vaporization/condensation temperatures may differ from the fridge. As mentioned earlier, the refrigerator can be defined as a heat pump that extracts heat from the foods placed in its compartment toward heating the kitchen. In fact, it should now make sense that heat extraction can be done from any source which is thermally coupled with evaporator. By using heat pump, we can extract heat from low-grade heat reserves in our environment such as sewage, rivers or sea [26]. Heat pumps are distinguished heat production alternatives since they obtain most of the required energy for heat generation from surrounding environments and hardly incorporate electricity. Consequently, the transferred heat is almost three to four times greater than the consumed power which indicates a significantly higher Coefficient of Performance (COP) value in comparison with electrical resistance heaters. There are mainly two types of heat pumps:

- ground source heat pump (GSHP): a heat pump that uses ground or any medium that is thermally coupled to the ground, such as groundwater, as a source of thermal energy.
- air-sourced heat pump: a heat pump that extracts energy directly from the surrounding air and there is no need to dig, drill or having large plot of land.

Although air-sourced heat pumps reduce the capital cost and are easy to install, they are not as efficient as GSHP due to air temperature fluctuations (low heat source at below zero temperature).

3.1.1 Heat pump efficiency

As previously noted, a heat pump transfers thermal energy from a low-temperature heat source to a high temperature heat sink, consuming electricity. The heat extracted from source side is upgraded to higher temperature and is employed for local heating. Assuming that energy loss in heat pump (refrigerant) cycle is negligible and all extracted and consumed energy is delivered to the end point, the total heat provided is given by

$$Q_{\mathrm{HP}} \approx Q_{\mathrm{E}} + W_{\mathrm{HP}}, \tag{3.1}$$

where W_{HP} is electrical energy used by heat pump. Q_{HP} and Q_{E} are heat provided by heat pump and heat extracted from heat source, respectively. The ratio of heat pump's heating provided to the electrical energy consumption is described by the coefficient of performance (COP) [20]

$$COP_{\mathrm{HP}} = \frac{Q_{\mathrm{HP}}}{W_{\mathrm{HP}}}. \tag{3.2}$$

The ideal heat pump process is represented by the reversed Carnot cycle [27]. The theoretical maximum heat pump efficiency depends on the temperature of the heat source (T_{source}) and the heat sink (T_{sink})

$$COP_{\max} = \frac{T_{\mathrm{sink}}}{T_{\mathrm{sink}} - T_{\mathrm{source}}}. \tag{3.3}$$

However, the maximum efficiency is not accessible in practice due to losses. Hence a grade of quality (η_{HP}) is used to demonstrate the thermal efficiency of heat pump

$$\eta_{\mathrm{HP}} = \frac{COP_{\mathrm{HP}}}{COP_{\max}}. \tag{3.4}$$

3.1.2 Thermal Model of Heat Pump

The heat pump scheduling approach proposed in this thesis aims to reduce the electricity bill while considering the thermal comfort of occupants in residential buildings. To guarantee the comfort, the pump scheduler should maintain the indoor temperature in a specific range that projects occupants' preferences. To maintain the indoor temperature, the scheduler needs to predict and track the indoor temperature changes over time with respect to the heat provided by the heat pump and the heat loss to the outdoor environment. The thermal model used in this study is derived from work of J. L. Cremer et al. [4] and M. Pau et al. [6]. The indoor temperature changes over the course of time are linked to heat provided by the heat pump and the heat loss to the outdoor environment, which can be observed through the energy balance equation

$$T_{\mathrm{IN}}^{t} = T_{\mathrm{IN}}^{t-1} + \frac{\Delta t}{\mu_{\mathrm{HS}}\gamma_{\mathrm{AR}}}(Q_{\mathrm{HP}}^{t} - Q_{\mathrm{LS}}^{t}), \tag{3.5}$$

where T_{IN} is the indoor temperature, Δt is the time step, μ_{HS} is the amount of house indoor air mass, γ_{AR} is air specific heat capacity, and Q_{HP}^{t} and Q_{LS}^{t} are heat provided by heat pump and lost from the house to the outdoor environment, respectively. The indoor air mass, μ_{HS} , is a parameter that depends on the size and geometrical characteristics of the house. In combination with the air heat capacity, γ_{AR} , it represents the thermal energy storage capacity of the house. The heat losses to the outdoor environment can be described as follows

$$Q_{\mathrm{LS}}^{t} = K_{\mathrm{HS}}(T_{\mathrm{IN}}^{t-1} - T_{\mathrm{OUT}}^{t-1}), \tag{3.6}$$

where K_{HS} is the heat loss factor. The heat generated by the heat pump is defined as

$$Q_{\mathrm{HP}}^{t} = \gamma_{\mathrm{AR}}\phi_{\mathrm{HP}}(T_{\mathrm{HP}} - T_{\mathrm{IN}}^{t-1}), \tag{3.7}$$

where T_{HP} is the temperature of the heat pump supply air (assumed as constant), and ϕ_{HP} is the air mass flow rate of the heat pump.

3.2 Metaheuristic Optimization Techniques

Optimization is the process of determining the most suitable solution for a specific problem. According to Kramer [28],"Optimization problems can be found in many fields, from natural sciences to math and computer science, from engineering to social and daily life. Whenever the task is to minimize an error, to minimize energy, weight, waste, effort or to maximize profit, outcome, success, and scores, we face optimization problems". Since the problems become more complicated during last decades, the necessity of new optimization techniques became evident [29]. In recent years, metaheuristic algorithms have been introduced and employed for solving many engineering optimization problems [30–32]. Such algorithms are attracting more and more interests because they [33]:

- can be implemented easily due to their uncomplicated concepts,
- have derivation-free mechanism,
- can avoid local optima,
- can be applied to various types of problems and disciplines.

Most metaheuristic algorithms have stochastic mechanisms [34]. The main characteristic of stochastic algorithms is randomness [35]. This means that these algorithms investigate entire search space for finding global optima using random operators. The randomized nature of metaheuristic techniques supports them to avoid local optima stagnation. However, it might make them unreliable, as stochastic optimization techniques may reach different results for a particular problem in each run due to their randomized behavior [36]. On the other hand, deterministic algorithms [37–39] are reliable, producing similar solutions for given problem with the same initial conditions. However, deterministic optimization approaches have highly potential for entrapping in local optima [40].

According to the No Free Lunch (NFL) theorem [41], there is no metaheuristic algorithm for solving all optimization problems. In other words, an optimizer may show outstanding performance on a specific problem, but the same algorithm may fail to solve a different set of problems.

In general, metaheuristics can be categorized into two main groups [42]: single-based and population-based methods. In single-solution-based methods the search process begins with one candidate solution that is improved over a number of iterations. In contrast, population-based techniques start the optimization process using a set of random solutions (a population), and this initial population is enhanced over the course of iterations. Population-based metaheuristics generally show better performance, since they benefit from the following advantages [42]:

- information-sharing about search space among candidate solutions
- local optima stagnation avoidance due to the support that each candidate receives from others
- great exploration capability

Considering the source of inspiration, metaheuristic algorithms can be classified as nature-inspired and human-behavior-inspired. Nature-inspired techniques deal with problems by imitating biological or physical phenomena. They can be divided into following groups [33]:

- evolution-based methods are inspired by the rules of natural evolution. The most popular evolution-inspired approach is the Genetic Algorithm (GA) [43] which simulates the Darwinian evolution. Other well-known techniques of this branch are Evolution Strategy (ES) [44], Genetic Programming (GP) [45] and Biogeography-Based Optimizer (BBO) [46].
- physics-based methods mimic the physical laws in the world. The most popular algorithms are Simulated Annealing (SA) [47, 48], Gravitational Search Algorithm (GSA) [49], Central Force Optimization (CFO) [50], Big-Bang Big-Crunch (BBBC) [51], and Galaxy-based Search Algorithm (GbSA) [52].
- swarm-based: In 1993, the Swarm Intelligence (SI) notion was presented for the first time [53]. According to Bonabeau et al. [54], SI is "The emergent collective intelligence of groups of simple agents". These techniques mimic the social

behaviour of natural colonies/packs. Some of the most popular SI techniques are Ant Colony Optimization (ACO) [55], Particle Swarm Optimization (PSO) [56] and, Artificial Bee colony (ABC) [57].

As mentioned earlier, there is another class of metaheuristic algorithms inspired by human behaviours, such as Tabu (Taboo) Search (TS) [58–60], Imperialist Competitive Algorithm (ICA) [61], Social-Based Algorithm (SBA) [62], Colliding Bodies Optimization (CBO) [63, 64], Firework Algorithm [65] and Group Counseling Optimization (GCO) algorithm [66, 67].

Regardless of the differences between metaheuristics, the common characteristic of these techniques is sorting out the optimization into exploration and exploitation [68–72]. The exploration phase makes agents to examine the search space as widely as possible. In this phase, random operators are required which enable the algorithm to cover the entire search space. On the other hand, exploitation phase makes individuals to search locally around the best acquired result found during exploration. The right balance between exploration and exploitation is a challenging task. However, it can play an important rule in increasing the chance of algorithm for converging to the global optima.

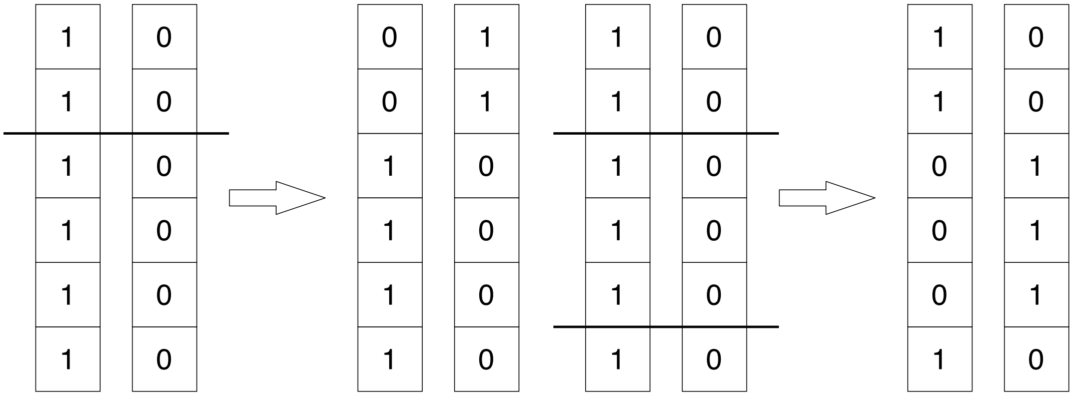
3.2.1 GA

The evolution concept, first proposed by Charles Darvin, describes biological advancement of species [73]. Species continue to live, especially despite hardships, due to their development and environmental adaption capability. Achievements of evolution in individuals survival are good reasons for adapting its fundamentals to solving optimization problems.

Genetic algorithm (GA) is an evolutionary algorithm inspired by the concepts of evolution in nature [43]. GA acts on a set of solutions (population) that represent candidate solutions to the optimization problem. The optimization process starts by randomly generating an initial population of solutions. In each iteration, genetic operators produce new solutions by considering the individuals in previous generation. Since well-performing individuals have relatively higher chance to participate in producing the new population, it is likely that quality of individuals improves from generation to generation. This gradual enhancement in the fitness value of agents may lead to global optima convergence. There are two elementary genetic operators: crossover and mutation.

3.2.1.1 Crossover

Crossover is an operator that allows the combination of the genes of two or more individuals [74]. In GA, the crossover is performed by swapping parts of two chromosomes (genotypes). A popular type of crossover is the n-point crossover, however, the single-point and double-point crossover operators illustrated in Fig. 3.2a and Fig. 3.2b are used more commonly. In a single-point crossover, parent chromosomes are split at a randomly chosen cut point and the parts are swapped to create two offspring. Similarly in double-point crossover the parent chromosomes are split at two randomly determined points. Multi-point crossover is accomplished by swapping multiple segments of one parent chromosome with the corresponding segments of another parent at random positions. [75].



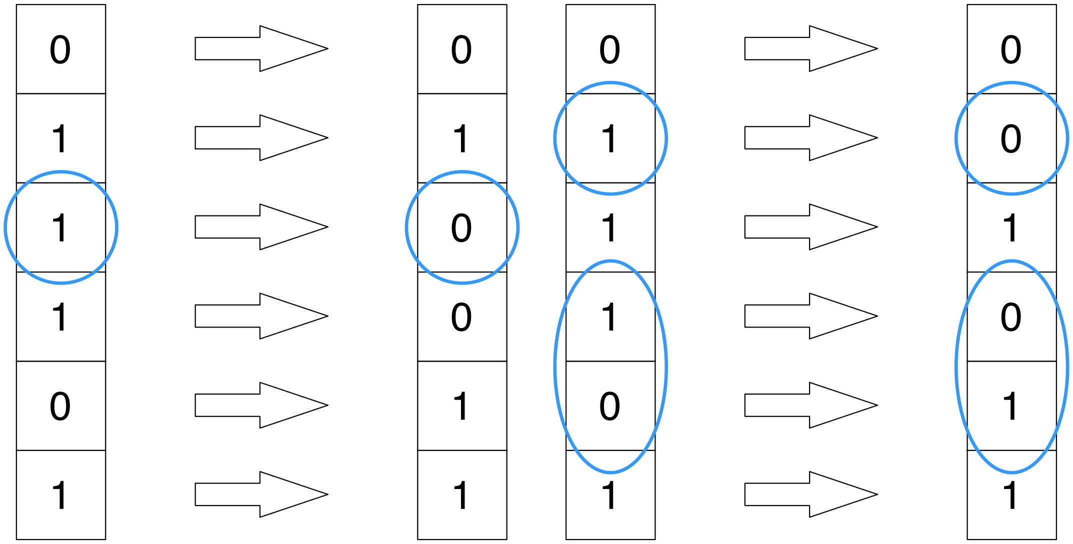
(a) Single-point

(b) Double-point

Figure 3.2: Crossover operators

3.2.1.2 Mutation

The second main operator in GA is mutation. It is an important method that enhances the diversity of candidate solutions by small, random changes. This can be implemented by randomly modifying the value of a gene in the parent chromosome. In binary coded chromosomes, bit flip mutation illustrated in Fig. 3.3 is usually used. It flips a zero bit to a one bit and vice versa with a defined probability, which plays the role of the mutation rate. Multi-point mutation operator is more common than single-point due to its higher diversity capability.



(a) Single-point (b) Multi-point

Figure 3.3: Mutation operators

3.2.1.3 Genotype-Phenotype Mapping

The new offspring population created by crossover and mutation operators has to be evaluated. The evaluation is performed considering the capability of each candidate solution to solve the optimization problem. In this regard, each coded chromosome (genotype) needs to be mapped to the actual solution (phenotype). “The process of translating genotypes into their corresponding phenotypes is called genotype-phenotype mapping” [76]. The genotype-phenotype mapping is usually not required in continuous optimization problems when the genotype is the solution itself.

3.2.1.4 Fitness

As mentioned earlier, the ability of all candidates in effectively dealing with a problem has to be evaluated. During the assessment process, the phenotype of a solution is assessed using a fitness (objective) function. The objective function evaluates the candidate solutions by measuring their quality, and a fitness value is assigned to each individual accordingly. Since the fitness function guides the search operations, it should be appropriately designed during the optimization modeling process. The function should be able to mitigate the negative effect of infeasible solutions generated by the optimization algorithm. Hence, the application of penalty mechanism for deteriorating the fitness of such impracticable solutions need to be considered [28].

The performance of GA and other metaheuristics is usually determined by counting the number of times the fitness function has been called until the global optima is found or the termination criterion is satisfied. Hence, minimizing the number of fitness function calls is very important, especially if function calls require a long running time.

3.2.1.5 Selection

To ensure the convergence toward the optimal solution, the best offspring should be selected to participate in the new population and play the parents role for the next generation. The selection process is based on the fitness values gained by each individual during the evaluation phase. Most selection methods are based on randomness; some well-known selection approaches are

- roulette wheel selection
- ranked selection
- tournament selection

Roulette wheel, also known as fitness proportional selection, selects individuals randomly with respect to their fitness values. Individuals' selection probabilities are calculated by normalizing their corresponding fitness value with the sum of all individuals' fitness values in the population. In ranked selection, the individuals are sorted and ranked based on their fitness values, while the worst individual ranks as one. The selection probability of individuals are determined by normalizing their rank with the population size. Tournament selection is another common method, where k individuals are randomly chosen from the population to compete with each other. The winner, i.e. the individual with the best fitness value, is selected and used as the new parent.

3.2.1.6 Termination

Due to the time and expense of fitness function assessments, running the optimization algorithm for an unlimited period is not feasible. Therefore, the processing time has to be restricted. The termination criterion determines when the optimization process terminates. Some common termination conditions are

- exceeding the maximum number of iterations,
- exceeding the maximum number of function evaluations (NFEs),
- achieving an acceptable solution,
- observing no significant progress after several iterations.

Please note that these criteria are not specific to GA and can be used in any optimization method.

3.2.2 PSO

Particle Swarm Optimization (PSO) was first proposed by Eberhart and Kennedy in 1995 [77, 78]. It is a type of swarm intelligence in which the social behaviour of flocking birds is simulated. When a swarm seeks for food, each individual moves around the environment (search space) independently [76]. Agents move randomly which increases their chance of finding food reservoirs. Since there is an information flow among swarm's individuals, when an individual finds food the other individuals will be informed respectively.

PSO investigates the problem space by regulating the pathway of particles. The swarm's movement includes two elements [75]: stochastic and deterministic. Each particle tends to move towards the global best position and its own best achievement so far, while preserving random movement propensity. The best location history of each particle (Pbest) is updated when the new solution dominates the current best history. In the same manner, the global best position (Gbest) is updated when a particle's best history is better than the current global best.

The position of each particle i is updated at each time step t, considering its current position x_i^t, current velocity v_i^t, personal best so far solution y_i^t, and the best so far solution of the entire swarm $\hat{y}$

$$v_i^{t+1} = v_i^t + c_1 r_1 (y_i - x_i^t) + c_2 r_2 (\hat{y} - x_i^t), \tag{3.8}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1}, \tag{3.9}$$

where c_1 and c_2 are positive acceleration coefficients, and r_1 and r_2 are random numbers between 0 and 1. The second and third parts of (3.8) are "cognition" and "social" factors, which represent the personal thinking of the agent and particles' collaboration respectively [79]. The effect of acceleration coefficients on algorithm's performance can be analyzed as follows [80]

- if $c_1 = 0$, the particles can communicate with each other, while they do not have cognitive abilities. Particles have the potential to fly to new regions in the search area however, they might entrap to local optima.

- if $c_2 = 0$, there is no information flow between particles and the swarm. Hence, PSO randomly investigates the problem space.

- if $c_1 = c_2 = 0$, the particles will preserve their initial velocity until they meet termination criterion.

In the initialization phase of PSO, the positions of all particles are randomly initialized while their early velocity is zero. The initial population plays an important role in reaching global optima. This is because the best history of each particle and the best history of entire swarm attract agents to search locally around the first group of candidates and they do not explore new areas in the search space [81]. The swarm's velocity helps to mitigate this problem, however, suitable boundaries need to be selected to guarantee the swarm convergence. A new parameter called inertia weight, w, was introduced by Shi and Eberhart to balance the exploration and exploitation [82, 83]. By applying the inertia weight, the new velocity update equation is as follows

$$v_i^{t+1} = wv_i^t + c_1r_1(y_i - x_i^t) + c_2r_2(\hat{y} - x_i^t). \tag{3.10}$$

To ensure the algorithm convergence, its exploration needs to be decreased along with increments of exploitation during the running time. Accordingly, the inertia weight is either linearly declined or determined using fuzzy system [84]. Decrements of inertia weight is suggested bounding from 0.9 to 0.4, which enhances the performance of PSO in several applications [85].

In 1999 Clerc indicated that controlling the particles' velocity and ensuring PSO convergence requires the use of constriction coefficient derived from eigenvalue analyses of swarm dynamics [86, 87]. Considering the constriction coefficient, the velocity update equation changes to [85]

$$v_i^{t+1} = \chi[v_i^t + \phi_1r_1(y_i - x_i^t) + \phi_2r_2(\hat{y} - x_i^t)], \tag{3.11}$$

where

$$\chi = \frac{2\kappa}{\mid 2 - \phi - \sqrt{\phi^2 - 4\phi} \mid}, \tag{3.12}$$

$$\phi = \phi_1 + \phi_2. \tag{3.13}$$

Here $\kappa \in [0,1]$ basically regulates the exploration and exploration. $\kappa \approx 0$ correlates with exploitation and fast convergence, while $\kappa \approx 1$ conforms to exploration and slow convergence. It has been proved that the convergence of PSO can be ensured if $\phi \geqslant 4$ is preserved [75].

3.2.2.1 BPSO

In general, many optimization problems are discrete binary in nature. Common examples incorporate determining the sequential of discrete components in arranging problems such as scheduling and routing tasks. A binary search space can be represented by a unit hypercube. Agents of binary algorithm can move between the corners of the hypercube by flipping groups of bits. Hence, some basic concepts of position updating process need to be modified in the binary version of PSO [88].

In continuous version of PSO, as demonstrated in previous section, the agents can fly all around the search space and position updating process can be implemented by adding updated velocity to current position of particle using (3.9). However, in binary space, due to dealing with 0 and 1 states, the position updating can not be implemented using aforementioned equation. Therefore, an approach is required to employ real domain velocities for updating the agents' position in binary space [89]. The idea for addressing this challenging problem is to use a transfer function to map velocity values with probability values to update the particles' position.

The binary version of PSO (BPSO) was proposed by Kennedy and Eberhart which enabled PSO to handle the binary search spaces [88]. In this approach a sigmoid function

$$S(v_{ik}^t) = \frac{1}{1+e^{-v_{ik}^t}}, \tag{3.14}$$

as a logistic transformation was employed to accomplish the mapping real values of velocities to probability values bounded to the interval [0,1]. The resulting position update is defined as follows [88]

$$\begin{aligned} \text{IF } (\text{rnd} < S(v_{ik}^{t+1})) \quad \text{THEN} \quad & x_{ik}^{t+1} = 1 \\ \text{ELSE} \quad & x_{ik}^{t+1} = 0 \end{aligned} \tag{3.15}$$

where rnd is a random number drawn from $[0, 1]$ using uniform distribution, v_{ik}^t is velocity of particle i at iteration t in dimension k, and x_{ik}^t represents the position of the particle i at iteration t in dimension k.

3.2.2.2 M-BPSO

To make sure that the BPSO algorithm can escape from local optima while maintaining fast convergence, a mutation operator has been added to the original BPSO. Before representing the employed mutation-based BPSO, it is necessary to briefly review the mutation concept and its application in metaheuristics. In biology, a mutation is an irrecoverable alteration in the deoxyribonucleic acid (DNA) sequence of a gene. They can change the amino acid sequence of the protein encoded by the gene which may result in discernible changes in the observable characteristics of an organism.

One of the main operators of GA is mutation, inspired by its biological concept. Similar to biological mutation, the characteristics of each agent can be altered by mutation operator. Generally, these changes are slight, however, they can improve the diversity of population and enhance the exploration capability of the algorithm [90]. Therefore, it can be expected that the performance of algorithm can be improved using a mutation operator. Considering the potential power of mutation and the fact that in some studies BPSO suffered from premature convergence and plunging into local optima [90, 91], the original algorithm has been modified by introducing mutation.

Chapter 4

Model Formulation and Setup

This chapter presents the settings and parameters' selection procedure for thermal modeling and optimization algorithms with respect to considered scenarios. The simulations are executed for two main scenarios, and several sub-scenarios. While the main scenarios determine consideration or negligence of thermal comfort delivered to occupants, sub-scenarios distinguish the time resolution of the simulations and static vs. dynamic determination of the maximum allowed continuous off-time of heat pump. These scenarios are classified as follows

- Optimization of HP schedule not considering occupant comfort
  - Optimization of HP with 30 minutes time interval resolution (N/30)
  - Optimization of HP with 15 minutes time interval resolution (N/15)
- Optimization of HP schedule considering occupant comfort with static off-time
  - Optimization of HP with 30 minutes time interval resolution (C/S/30)
  - Optimization of HP with 15 minutes time interval resolution (C/S/15)
- Optimization of HP schedule considering occupant comfort with dynamic off-time
  - Optimization of HP with 30 minutes time interval resolution (C/D/30)
  - Optimization of HP with 15 minutes time interval resolution (C/D/15)

It should be mentioned that the constraint regarding the maximum continuous off-time of heat pump is applied for considering the thermal comfort of occupant. Hence, in the case that thermal comfort is not considered, the sub-scenarios regarding the static and dynamic off-time approaches are not applicable.

The considered time horizon for the heat pump scheduling is one day. The initial time of the scheduling problem is midnight and the day is separated to 48 or 96 time slots with respect to the considered scenario, resulting in a time step of Δt=30 min or Δt=15 min, respectively.

4.1 Thermal Model of the Building

This section provides details about the parameters of the simulated model in particular building with respect to heat pump thermal model introduced in the background chapter. The thermal dynamics of the building are simulated based on the introduced model and provided parameters. The simulations are executed considering two different scenarios: the 15 minutes and 30 minutes time interval resolution.

4.1.1 Parameters of the Thermal Model

It is assumed that the house is equipped with air source heat pump as space heater. The parameters of considered heat pump, shown in Table 4.1, are derived from De Angelis et al. [23]. According to the considered scenarios the minimal time period the heat pump must run could be 30 min or 15 min, corresponding to one time interval in each introduced scenario.

Table 4.1: Heat pump parameters

Air flow rate	**Power**	**Supply air temperature**
1148 Kg/h	2080 W	30°C

The indoor air mass and the heat loss factor are calculated based on the geometric dimensions of the house after De Angelis et al. [23] and J. Cremer et al. [4]. The house geometry and parameters are presented in Table 4.2. Based on the parameters, the heat loss factor is calculated as follows [4]

$$\begin{aligned} K_{\text{HS}} &= \nu_{WA}(2(L+W)H - N_{\text{WI}}A_{\text{WI}}) + N_{\text{WI}}\nu_{\text{WI}}A_{\text{WI}} \\ &= 191.16 \text{ kJ/h°C} \end{aligned} \tag{4.1}$$

Using the air density at standard conditions (ρ_{AR}=1.2041 kg/m^3) the indoor air mass is calculated by following equation [4]

$$\begin{aligned} \mu_{\text{HS}} &= \rho_{\text{AR}}(LWH + 0.25LW^2\tan(\beta)) \\ &= 3946 \text{ kg} \end{aligned} \tag{4.2}$$

Table 4.2: House geometry and parameters

Parameters	**Value**
House length (L)	20 m
House width (W)	20 m
House height (H)	4 m
Roof pitch (β)	40°
Number of windows (N_{WI})	6
Area of each window (A_{WI})	1 m^2
Thermal transmittance for walls (ν_{WA})	0.15 W/m^2K
Thermal transmittance for windows (ν_{WI})	1 W/m^2K

The reference indoor temperature is considered $T_{\text{IN}}^{\text{ref}}$=21°C. As starting point for the simulation, the initial indoor temperature is assumed to be equal to reference temperature, $T_{\text{IN}}^{\text{init}}$=$T_{\text{IN}}^{\text{ref}}$. It should be mentioned that two-degree of freedom is assigned to scheduler which will set the minimum acceptable value for indoor temperature to 19°C.

To guarantee thermal comfort of the occupants, two different methodologies about the maximum allowed continues off-time of heat pump are considered,

- static off-time method
- dynamic off-time method

In the static off-time method, a constant value for maximum allowed continuous off-time of heat pump is defined, and its violation is penalized. According to the assumption about reference temperature (21°C) and the minimum acceptable temperature (19°C), the maximum continuous off-time is calculated utilizing following equation derived from M. Pau et al. [6]

$$\Delta t_{\max} = \frac{\mu_{\mathrm{HS}} \gamma_{\mathrm{AR}}}{K_{\mathrm{HS}}} \frac{\Delta T_{\mathrm{dof}}}{(T_{\mathrm{IN}}^{ref} - T_{\mathrm{OUT}}^{min})}, \tag{4.3}$$

where ΔT_{dof} is the degree of freedom for indoor temperature and T_{OUT}^{min} presents the minimum outdoor temperature during optimization progress. By considering the air heat capacity γ_{AR}=1.005 (kj/kg°C) and using the provided values for other parameters, the maximum continuous off-time of heat pump is calculated as $\Delta t_{\max}$=1.56 h, corresponding to 3 or 6 time slots when the scheduling time interval resolution is 30 min or 15 min, respectively. The maximum continuous off-time duration of the heat pump in this method is fixed and not updated during the entire scheduling period. It is expected that this method does not provide sufficient flexibility for the algorithm to benefit from the maximum thermal storage capacity of the building.

In the dynamic off-time method, the maximum allowed continuous off-time of heat pump is updated at each time interval after calculation of new indoor temperature based on Eq. 3.5. Given the indoor and outdoor temperature, the maximum continuous off-time of heat pump can be calculated as follows

$$\Delta t_{\max} = \frac{\mu_{\mathrm{HS}} \gamma_{\mathrm{AR}}}{K_{\mathrm{HS}}} \frac{(T_{\mathrm{IN}}^{t} - T_{\mathrm{IN}}^{min})}{(T_{\mathrm{IN}}^{t} - T_{\mathrm{OUT}}^{t})}, \tag{4.4}$$

where T_{IN}^{min} is the minimum acceptable indoor temperature set to 19°C. While this approach provides a good approximation of the maximum off-time of heat pump, it may lead to some inaccuracy in the cases when the outdoor temperature changes frequently. In addition, it puts heavy computation load on the optimization algorithm, slowing down the optimizer. To address the aforementioned issues while receiving dynamic off-time advantages, the minimum indoor temperature constraint takes the place of maximum continuous off-time limitation in performed simulations. The new constraint allows the heat pump to be off while the indoor temperature is higher than the minimum set point, which provides more flexibility for the algorithm to benefit from the maximum thermal storage capacity of the building. The minimum allowed indoor temperature is set by the occupant and its violation is penalized.

As electricity price signal profile, the winter time-of-use (TOU) price set by the Ontario Energy Board (OEB), shown in Table 4.3 [92], is used.

Table 4.3: Time-of-use price of electricity

TOU periods	**TOU prices**
19:00-7:00	0.101 $/kWh
11:00-17:00	0.144 $/kWh
07:00-11:00, 17:00-19:00	0.208 $/kWh

The outdoor temperature profile during the optimization period is illustrated in Fig. 4.1.

4.1.2 Simulation of Thermal Dynamics of the Building

As previously mentioned, the thermal storage capacity of building is the main factor for scheduling the heat pump. The first step is the prediction of daily heating energy demand in the house with respect to the indoor reference temperature, heat losses to the outdoor environment, and heat gains. Given the heating energy demand, required heat pump running time can be calculated using the model presented in section 3.1.2.

This thesis is concerned with intelligent heat pump scheduling approach to minimize the cost of electricity, while maintaining the occupant comfort in terms of indoor temperature. The schedule of other appliances, and thus the contribution of internal heat gain to the heating energy demand, is not taken into account. Hence, the only

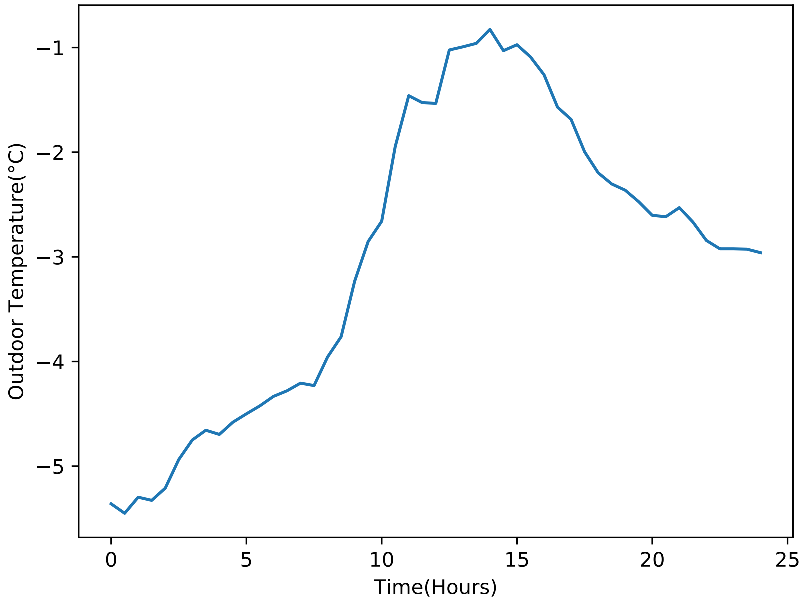


Figure 4.1: Outdoor temperature profile

heat source considered is the heat pump. M. Z. Degefa et al. [93] indicated that internal heat gain varies between 5% and 7% from the daily heating energy demand in typical single-family detached households on a cold winter day. Therefore, neglecting the internal heat gain does not have a significant effect on the accuracy of the implemented simulations.

Before starting the heat pump scheduling simulations, it is necessary to simulate the thermal dynamics of the house so that the indoor temperature variations can be tracked. The thermal dynamics of the building are simulated based on the introduced model and parameters while considering two different scenarios, presented in the following sections.

4.1.2.1 Simulation results based on 30 minutes time interval resolution

According to the outdoor temperature profile, thermal model and parameters setting provided earlier, the indoor temperature variations while the heat pump is kept off during the whole optimization period (one day) is simulated and presented in Fig.

4.2. When the heat pump is turned off, its heat flow, Q^t_{HP}=0. Hence, the temperature update equation (3.5) becomes

$$T^t_{\mathrm{IN}} = T^{t-1}_{\mathrm{IN}} - \frac{\Delta t}{\mu_{\mathrm{HS}}\gamma_{\mathrm{AR}}}(Q^t_{\mathrm{LS}}). \tag{4.5}$$

The simulation results show that the indoor temperature drops to 4.64°C at the end of the day from its initial value of 21°C. These slow thermal dynamics are due to the storage capability of the building, which leads to the possibilities to interrupt the heat pump operation.

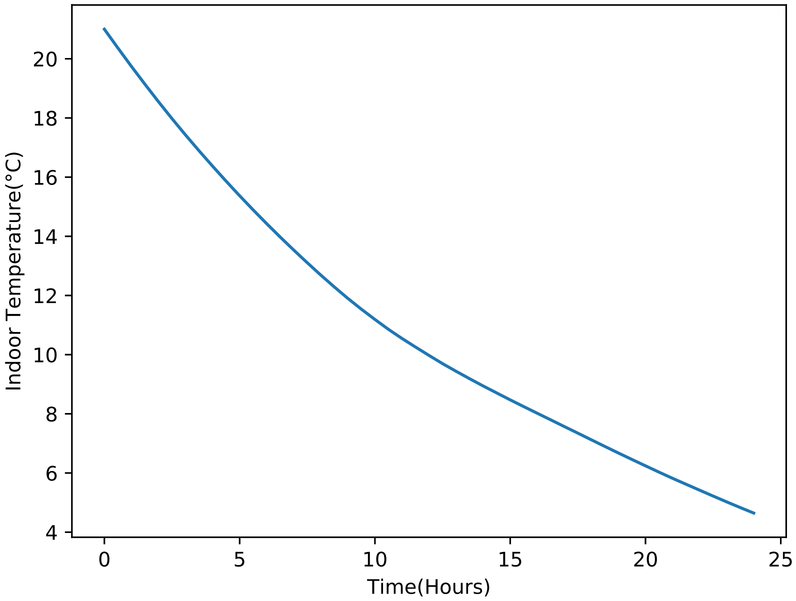


Figure 4.2: Indoor temperature variation with 30 min time resolution while HP is turned off

4.1.2.2 Simulation results based on 15 minutes time interval resolution

When the time interval resolution is increased to 15 minutes, the outdoor temperature sampling resolution needs to be updated. Hence, every 15 minutes average of outdoor temperature profile shown in Fig. 4.1 is considered as the outdoor temperature value during each time interval. Accordingly, the indoor temperature variation while heat pump is kept off during the whole simulation period is updated and presented in Fig. 4.3. The simulation results show that the indoor temperature drops to 4.70°C at the end of the day while its initial value is 21°C. In comparison with prior scenario in which the time interval resolution was 30 minutes, the final indoor temperature increased by about 0.04°C which is due to the increase of outdoor temperature resolution.

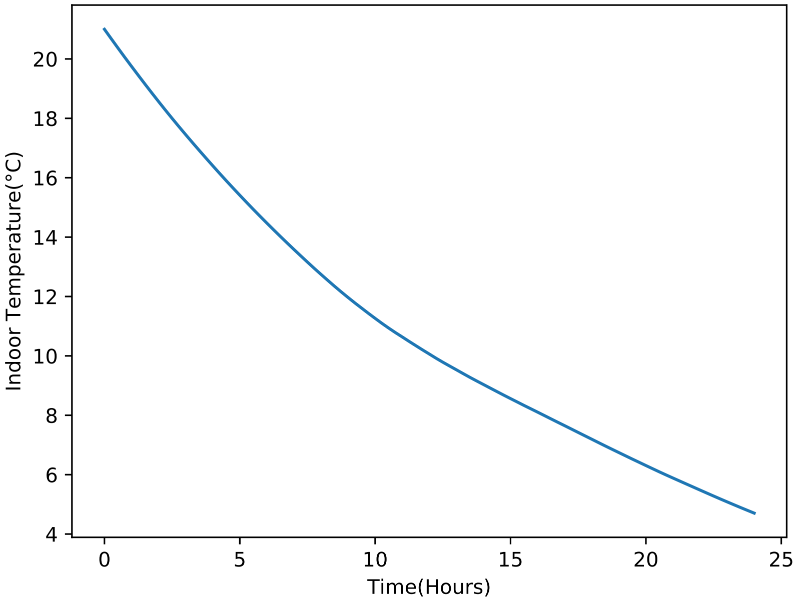


Figure 4.3: Indoor temperature variation with 15 min time resolution while HP is turned off

4.2 Objective Function and Constrains

The goal of the scheduler is to minimize the cost of electricity by appropriately engaging the heat pump based on the time variable electricity price signal (time of use, TOU). The corresponding objective function can be formulated as follows

$$f = \sum_{t=1}^{N_{\text{intvl}}} HP_{\text{status}}^{t} \cdot EPS(t) \cdot HP_{\text{power}} \cdot \Delta t, \tag{4.6}$$

where HP_{status} represents the on/off status of the heat pump during time period Δt, EPS is electricity price signal during that time period, HP_{power} is the heat pump power, and N_{intvl} is number of time intervals. As mentioned earlier, the optimization period (a day) is divided to Δt=30/15-minute periods and thus encompasses 48/96 intervals, respectively.

In this scheduling problem, there are several time slots with the same electricity price. Therefore, permuting corresponding segments of the optimization time period does not affect the obtained cost but may change the indoor temperature profile. To facilitate optimization for the secondary objective (highest minimum temperature) for given cost, the standard search procedures of the examined algorithms has been augmented. This is accomplished through an additional temperature attribute evaluated for each candidate solution using (3.5). Since the primary objective is the cost minimization, the temperature attribute is only considered for comparable candidate solutions: the individual with the highest minimum temperature will outperform the other candidates with the same cost.

To reduce the overall cost of electricity consumption, the scheduler shifts the heat pump working times from periods of high electricity price to times with low prices. However, this may affect the indoor temperature and thus the thermal comfort of the occupants. To ensure that the indoor temperature is always within the acceptable range, the continuous off-time of the heat pump should be limited. This can be enforced through a constraint that penalizes the scheduler when a maximum allowed continuous off-time is exceeded. Another constraint is added to control the minimum run-time of the heat pump to avoid unnecessary off-on/on-off cycling that would decrease the useful lifetime of the pump. In this thesis, the minimum run time of the heat pump is considered equal to one time step; this way the minimum time constraint is always satisfied and its satisfaction does not have to enforced. As presented in section 4.1.1, there are two different approaches to determine the maximum continuous off-time: static and dynamic. In the static case, the penalty is determined from the deviations of the actual continuous off-time and the maximum allowed continues off-time (HP_{MCOT}) as follows

$$\begin{aligned} \text{IF } (\sum_{t=1}^{t+HP_{\text{MCOT}}} HP_{\text{status}}^{t} = 0) \text{ THEN } \quad & OT_{\text{dev}}^{t} = 1 \\ \text{ELSE } \quad & OT_{\text{dev}}^{t} = 0. \end{aligned} \tag{4.7}$$

After multiplying by a penalty factor, F_{penalty}, the entire term,

$$F_{\text{penalty}} \sum_{t=1}^{N_{\text{intvl}}-HP_{\text{MCOT}}} OT_{\text{dev}}^{t} \tag{4.8}$$

is added to the original cost function (4.6). According to (4.3), the HP_{MCOT}=$3\Delta t$ for the 30 minutes time resolution and HP_{MCOT}=$6\Delta t$ for the 15 minutes resolution. In the dynamic off-time scenario, since the constraint is applied with respect to the minimum temperature set point determined by the occupant (considered to be 19°C), the penalty term is determined from the deviations of the actual indoor temperature and the minimum acceptable indoor temperature as follows

$$\begin{aligned} \text{IF } (T_{\text{IN}}^{t} < T_{\text{IN}}^{min}) \quad \text{THEN} \quad & TMP_{\text{dev}}^{t} = T_{\text{IN}}^{min} - T_{\text{IN}}^{t} \\ \text{ELSE} \quad & TMP_{\text{dev}}^{t} = 0, \end{aligned} \tag{4.9}$$

after multiplying by a penalty factor F_{penalty}, the entire term,

$$F_{\text{penalty}} \sum_{t=1}^{N_{\text{intvl}}} TMP_{\text{dev}}^{t} \tag{4.10}$$

is added to standard objective function (4.6).

4.3 Optimization Algorithms Setup

As previously mentioned, this thesis presents an intelligent approach to heat pump scheduling problem based on metaheuristic optimization algorithms. In particular, the genetic algorithm and binary particle swarm optimization are employed in this study. Background of the metaheuristic optimization techniques including the employed algorithms is provided in the third chapter of this document.

As mentioned earlier, the original version of binary particle swarm optimization suffers from local optima stagnation. This thesis aims to mitigate this issue, by enhancing the basic version of BPSO in two steps. In the first step, the multi-point mutation operator is added to the original BPSO with s-shaped transfer function. Because the performance of M-BPSO-S is still not acceptable in terms of speed of convergence, in the second step the s-shaped transfer function is replaced by v-shape transfer function, resulting in M-BPSO-V algorithm.

More implementation details of each optimization algorithm with respect to the heat pump scheduling problem are provided in the following sections.

4.3.1 GA

As stated in section 3.2, the GA optimization process starts by generating a group of random individuals, called the initial population. All individuals are evaluated using the fitness function, and a fitness value is assigned to each individual accordingly. Individuals with better fitness values (lower electricity cost in our case) are selected to participate in new population generation. New population is created using crossover and mutation operators described in Chapter 3. This generation/evaluation cycle continues until a termination criterion is satisfied. Exceeding the maximum number of iterations is considered as termination condition in this project. Parameter settings for implementing the GA in this investigation are listed in Table 4.4. The crossover and mutation percentages determine the number of individuals in the population affected by crossover and mutation operators, respectively. Since in our study the chromosomes are coded in binary format, the mutation rate represents the number of randomly selected genes whose value should be modified. P-SPX and P-DPX show the probability of single point and double point approach as the crossover operator, respectively. The parameters of the algorithm, such as iteration budget, population size, crossover percentage, etc. have been selected considering the problem complexity. Another factor considered in the selection of GA's parameters is the NFE budget. In this regard, to provide a fair comparison of the algorithm performance, the GA's parameters have been selected so that their NFE budgets are balanced.

Table 4.4: GA parameters' setting

Parameters	**Values**	
	30min resolution	**15min resolution**
Number of dimensions	48	96
Population size	500	500
Maximum No. of iterations	500	500
F_{penalty}	100	100
Crossover percentage	0.8	0.8
Mutation percentage	0.4	0.4
Mutation rate	0.04	0.02
P-SPX	0.2	0.2
P-DPX	0.8	0.8

4.3.2 M-BPSO-S

In BPSO, the optimization process starts by randomly generating an initial population of particles (candidate solutions). Each particle has four attributes including position, velocity, fitness and personal best history, all updated every iteration. The velocity of particles in each dimension is updated using (3.10). Given the updated velocities the positions of particles are updated considering the sigmoid transfer function and the position updating rules stated (3.15). New position of each particle is evaluated using fitness (objective) function. In the case that newly obtained fitness value of a particle dominates its personal best record, the particle's best-so-far solution is updated and the best-so-far solution of the entire swarm is checked for a possible update. Otherwise, no updates regarding the particle's best record and the swarm's best record are performed.

In the proposed mutation based BPSO, after updating the velocity and position of all particles in each iteration using the original BPSO principles, a mutation operator is applied to randomly selected parents and resulting offspring are evaluated. To enhance optimization capability, three strategies regarding mutation application are considered

- multi-point mutation,
- dynamic mutation rate,
- successful mutations acceptance.

The multi-point mutation described in Fig. 4.4 is used in this work since it maintains more diversity in comparison to a single point mutation. The percentage of the population that is affected by mutation operator is kept constant, while the gene's mutation rate is linearly decreased during the optimization process. This results in preserving the exploration decrements along with increments of exploitation during the optimization process which significantly improves the performance of the algorithm. If mutants outperform their parents, they take their parents' place in the swarm. Otherwise, the population does not change.

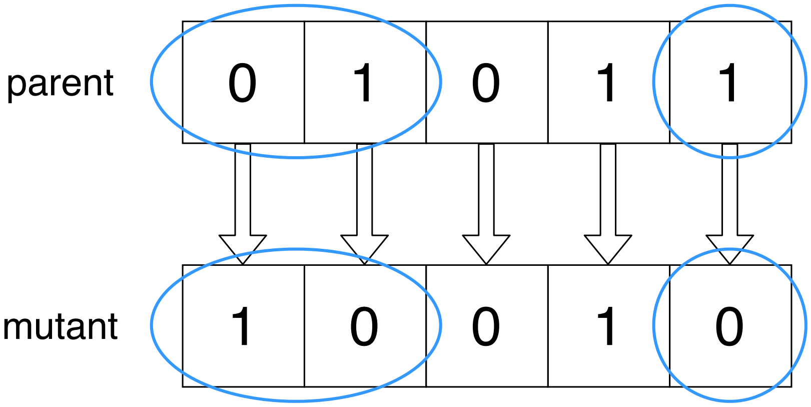


Figure 4.4: Multi-point mutation in M-BPSO

Since the parameter tuning can significantly improve the algorithm performance, we use constriction coefficient method for determining the acceleration coefficients, c_1 and c_2, as described in chapter 3. In addition, to ensure convergence, the inertia weight and mutation rate are linearly decreased during the optimization process. This results in gradual from exploration to exploitation. Based on these considerations, the velocity equation is updated as follows

$$v_i^{t+1} = (w_{\mathrm{damp}})^{it-1} w v_i^t + c_1 r_1 (y_i - x_i^t) + c_2 r_2 (\hat{y} - x_i^t), \tag{4.11}$$

where $w{=}\chi$ as defined by (3.12) and (3.13), and

$$c_1 = \chi \phi_1, \tag{4.12}$$

$$c_2 = \chi \phi_2. \tag{4.13}$$

The damping factor, w_{damp}, of inertia weight is raised to the power of it, the current iteration number. There is also the damping factor of mutation rate, MR_{damp}, applied using the following rule

$$\begin{aligned} \text{IF } (it \leqslant 21) \quad \text{THEN} \quad & MR = MR_{\mathrm{init}} (MR_{\mathrm{damp}})^{it-1} \\ \text{ELSE} \quad & MR = MR^{it=20} \end{aligned} \tag{4.14}$$

where MR_{init} is the initial value of mutation rate. All parameters used for the implementation of M-BPSO-S are summarized in Table 4.5. The parameter selection has been made with respect to the problem complexity and to balance the NFE of all three examined algorithms.

4.3.3 M-BPSO-V

The most important part of BPSO is the transfer function which maps a continuous search space to a discrete binary space. In other words, the probability of altering the particle's position (flipping a zero bit to 1 or vice versa) in each dimension is

Table 4.5: M-BPSO-S parameters' setting

Parameters	**Values**	
	30min resolution	**15min resolution**
Number of dimensions	48	96
Population size	500	500
Maximum No. of iterations	500	500
F_{penalty}	100	100
ϕ_1	2.05	2.05
ϕ_2	2.05	2.05
ϕ	4.1	4.1
κ	1	1
χ	0.7298	0.7298
w	0.7298	0.7298
w_{damp}	0.9975	0.9975
c_1	1.49	1.49
c_2	1.49	1.49
Mutation percentage	0.2	0.2
MR_{init}	0.083	0.041
MR_{damp}	0.965	0.965

determined by the transfer function [89]. According to E. Rashedi et al. [94], this function should be selected considering the following criteria

- The transfer function's range should be limited to [0,1] interval, because it presents the probability that a particle needs to update its position.

- A transfer function should provide a high position updating probability in the case that the absolute value of particle's velocity is large. Since particles with large absolute value of velocity are far from the best solution, they need to update their position with a high probability.

- A transfer function should also present a low position updating probability, in the case that the absolute value of the particle's velocity is small.
- The output value of a transfer function should increase along with velocity increments. Particles having great velocity are going far away from the best solution, so they should change their position vector in order to go back to their prior positions.
- The output value of transfer function also should decrease along with velocity decrements.

The transfer function with the aforementioned features is a good choice for mapping the continuous search space to discrete space while maintaining the original searching principles of the optimization algorithm, such as Gbest and Pbest concepts in PSO.

S. Mirjalili et al. [89] proposed a modified BPSO algorithm based on v-shaped transfer function with different position updating rules. The authors introduced six new transfer functions classified into two families, s-shaped and v-shaped. Evaluation results indicated that v-shaped family of transfer functions with respect to their own position updating method can enhance the performance of the original BPSO in both local optima avoidance and convergence rate aspects. Therefore, this study uses the v-shape transfer function to improve the algorithm performance. In this regard, new transfer function and position updating rules are considered as follows [89]

$$T(v_{ik}^t) = \left| \frac{v_{ik}^t}{\sqrt{1 + (v_{ik}^t)^2}} \right|, \tag{4.15}$$

$$\begin{aligned} \text{IF } (\text{rnd} < T(v_{ik}^{t+1})) \quad \text{THEN} \quad & x_{ik}^{t+1} = (x_{ik}^t)^{-1} \\ \text{ELSE} \quad & x_{ik}^{t+1} = x_{ik}^t \end{aligned} \tag{4.16}$$

where, $(x_{ik}^t)^{-1}$ is the complement of x_{ik}^t. The advantage of this method is that v-shaped transfer functions do not force particles to take 0 or 1 values. However, they encourage particles to keep their current positions when their velocity values are low or switch to their complements when their velocity values are high.

Otherwise, the M-BPSO-V keeps the same configuration as M-BPSO-S, while its transfer function and position updating rules are superseded by (4.15) and (4.16), respectively.

Chapter 5

Simulation Results and Analysis

According to the model setup and related constraints for preserving the thermal comfort of the occupant, the heat pump schedule has been determined for the following scenarios introduced in prior chapter:

- Optimization of HP schedule not considering occupant comfort
  - Optimization of HP with 30 minutes time interval resolution (N/30)
  - Optimization of HP with 15 minutes time interval resolution (N/15)
- Optimization of HP schedule considering occupant comfort with static off-time
  - Optimization of HP with 30 minutes time interval resolution (C/S/30)
  - Optimization of HP with 15 minutes time interval resolution (C/S/15)
- Optimization of HP schedule considering occupant comfort with dynamic off-time
  - Optimization of HP with 30 minutes time interval resolution (C/D/30)
  - Optimization of HP with 15 minutes time interval resolution (C/D/15)

This chapter presents the scheduling results obtained using GA, M-BPSO-S and M-BPSO-V with respect to these scenarios. All algorithms are tuned and use the same configuration in terms of population size, iteration budget and number of function evaluations (NFEs), as detailed in chapter 4. All algorithms are executed using the same computing resources. To allow a fair comparison. All scheduling simulations were executed 50 times and the best results are reported and analyzed. In addition, details on distribution of all 50 simulation results are also presented. So that their variability can be examined as a proxy for consistency of the individual algorithms.

5.1 Optimization of HP Schedule Not Considering Occupant Comfort

In this scenario, the thermal comfort of occupants is not considered. Therefore, the maximum allowed continuous off-time of heat pump is not relevant and thus not considered as a constraint.

5.1.1 Optimization of HP with 30 minutes time interval resolution

In this scenario the time resolution is set to 30 minutes. Accordingly, the minimum running time of the heat pump is 30 minutes.

To simulate the heat pump scheduling, in addition to all previously presented parameters, the required heat pump running time during the optimization period needs to be determined. Since the heat pump should compensate the heat losses to the outdoor environment for keeping the indoor temperature close to the reference set point, finding the heat loss amount is the main step toward the determination of heat pump running time. The amount of heat loss while satisfying the indoor reference temperature is estimated by substituting the $T_{\mathrm{IN}}^{\mathrm{ref}}$ in (3.6), resulting in the following expression

$$Q_{\mathrm{LS}}^{t} = K_{\mathrm{HS}}(T_{\mathrm{IN}}^{ref} - T_{\mathrm{OUT}}^{t-1}). \tag{5.1}$$

Given the heat loss flow in each time interval, the total amount of heat loss during the optimization period is calculated as follows

$$\begin{aligned} Q_{\text{LS}}^{total} =& \textstyle\sum_{t=1}^{48} Q_{\text{LS}}^{t} \Delta t \\ =& \ 110.103MJ. \end{aligned} \tag{5.2}$$

As previously mentioned, the heat pump should satisfy the heat demand to maintain the indoor temperature close to the reference temperature set point. The heat generation flow of the pump during the optimization period can be calculated by substituting the $T_{\text{IN}}^{\text{ref}}$ in (3.7) as follows

$$\begin{aligned} Q_{\text{HP}} &= \gamma_{\text{AR}} \phi_{\text{HP}} (T_{\text{HP}} - T_{\text{IN}}^{ref}) \\ &= 10.383MJ/h. \end{aligned} \tag{5.3}$$

Given the heat flow of the pump together with heat demand during the optimization period, the required heat pump running time (HP_{RRT}) can be calculated as follows

$$\begin{aligned} HP_{\text{RRT}} &= \frac{Q_{\text{LS}}^{total}}{Q_{\text{HP}}} \\ &= 10.60hrs, \end{aligned} \tag{5.4}$$

corresponding to 22 time slots based on considered time interval resolution and minimum running time of the heat pump. To ensure that the required running time of the heat pump (HP_{RRT}) is satisfied during the scheduling progress, a new constraint regarding the required heat pump running time is considered and its violation is penalized. The penalty is determined from the deviation of the running time of the heat pump from its prescribed limit. The following term is added to original objective function (4.6).

$$F_{\text{penalty}}(\mid HP_{\text{RRT}} - \sum_{t=1}^{N_{\text{intvl}}} HP_{\text{status}}^{t} \mid) \tag{5.5}$$

The result of the scheduling simulations using the three selected optimization algorithms are provided in the following sections.

5.1.1.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.1. As a reminder, the indoor temperature dynamics is simulated and updated using (3.5), (3.6) and (3.7). Since in this case the algorithm only seeks to minimize the cost, the heat pump is kept continuously off for a long time, especially during the time with relatively higher electricity prices.

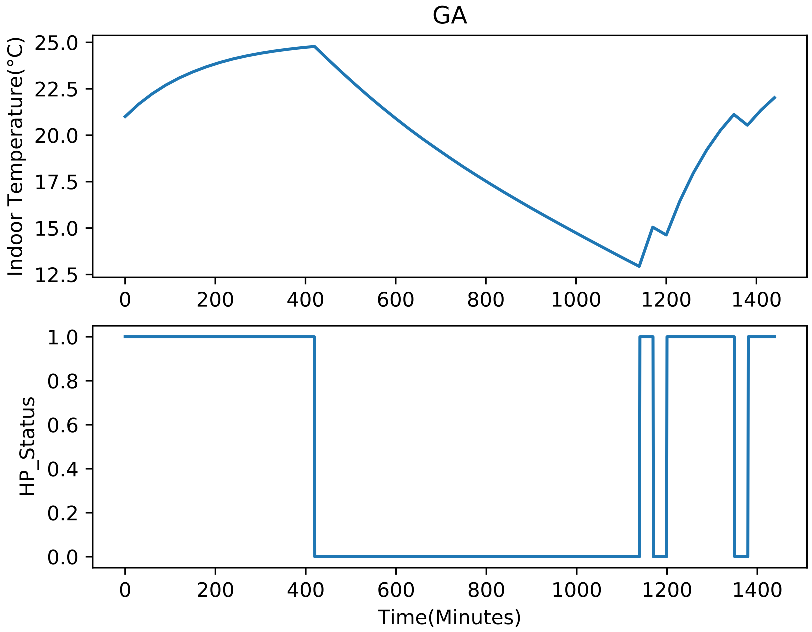


Figure 5.1: Heat pump schedule and corresponding indoor temperature (N/30 GA)

Fig. 5.2 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. It is clear that the cost minimization leads to a significant temperature drop which has a negative effect on the thermal comfort of the occupants. The variation of minimum indoor temperature versus electricity cost during the optimization progress is shown in Fig. 5.3.

As previously explained, the HP scheduling simulations were executed 50 times and the best results achieved by each algorithm are reported. In this regard, the performance of GA in terms of the best-obtained cost and minimum indoor temperature over 50 execution times are shown in Fig. 5.4a and Fig. 5.5a, respectively. The graphs show that GA reached the global optima point in all runs which shows its consistency in solving the problem. The same is confirmed by the histograms in Fig.

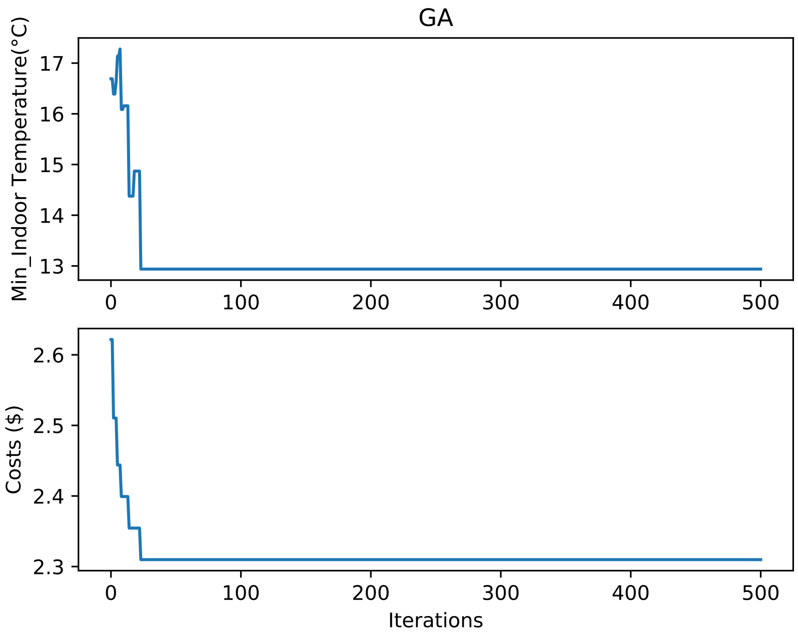


Figure 5.2: Minimum indoor temperature and cost variation (N/30 GA)

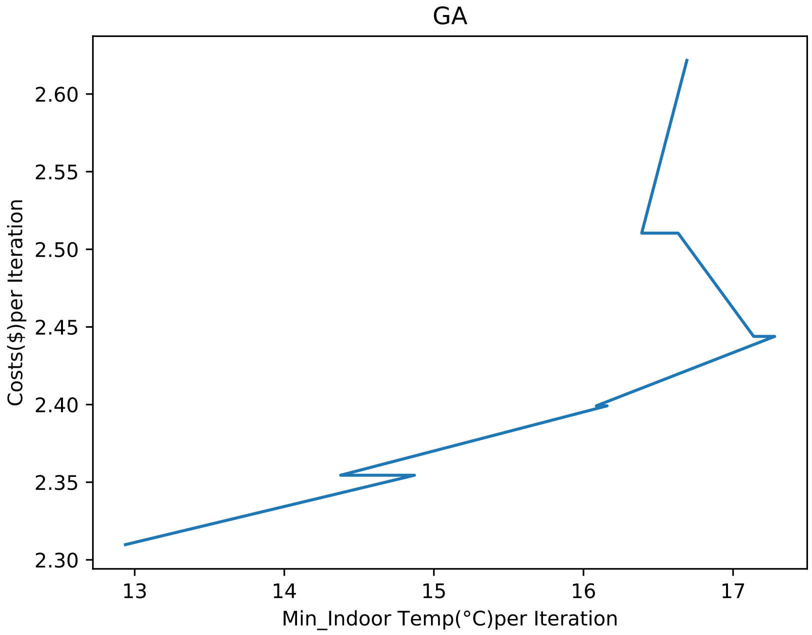
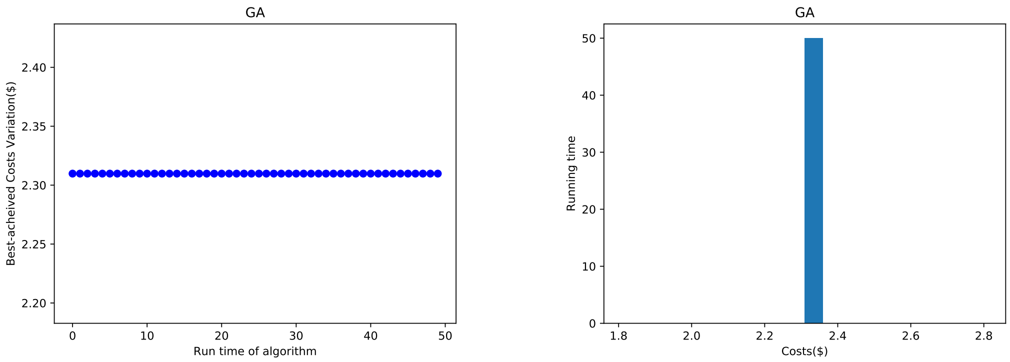


Figure 5.3: Search space trajectory of cost vs. minimum indoor temperature (N/30 GA)

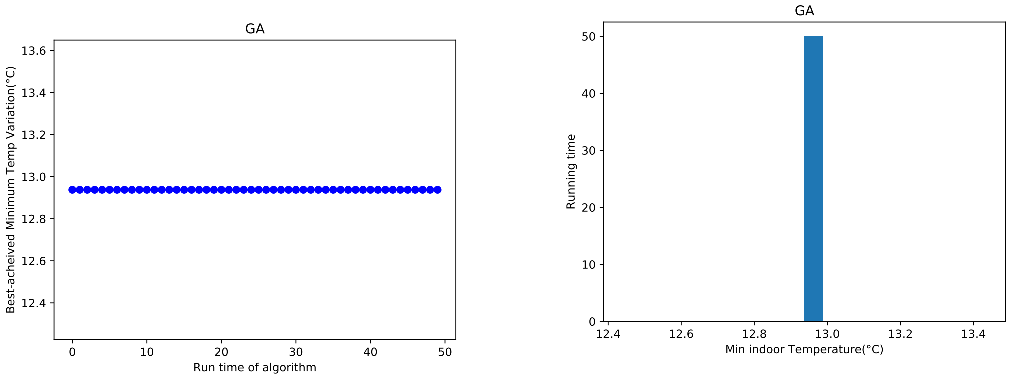
5.4b and Fig. 5.5b.



(a) Costs

(b) Histogram of the costs

Figure 5.4: The best achieved costs during each run (N/30 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.5: The best achieved min temp during each run (N/30 GA)

5.1.1.2 Results using M-BPSO-S

The heat pump running schedule and indoor temperature corresponding to the best result found by M-BPSO-S are shown in Fig. 5.6. The optimized schedule is nearly the same as the schedule found using the GA, keeping the heat pump continuously off for a long period to minimize the overall energy cost. However, M-BPSO-S did not converge to the global optima point.

The variation of minimum indoor temperature and cost during the optimization progress is illustrated in Fig. 5.7. While GA converged to global optimum cost and corresponding indoor temperature, the M-BPSO-S algorithm has not converged to the optimum point during the optimization progress in allotted number of iterations. More information about minimum indoor temperature changes versus electricity cost is provided in Fig. 5.8.

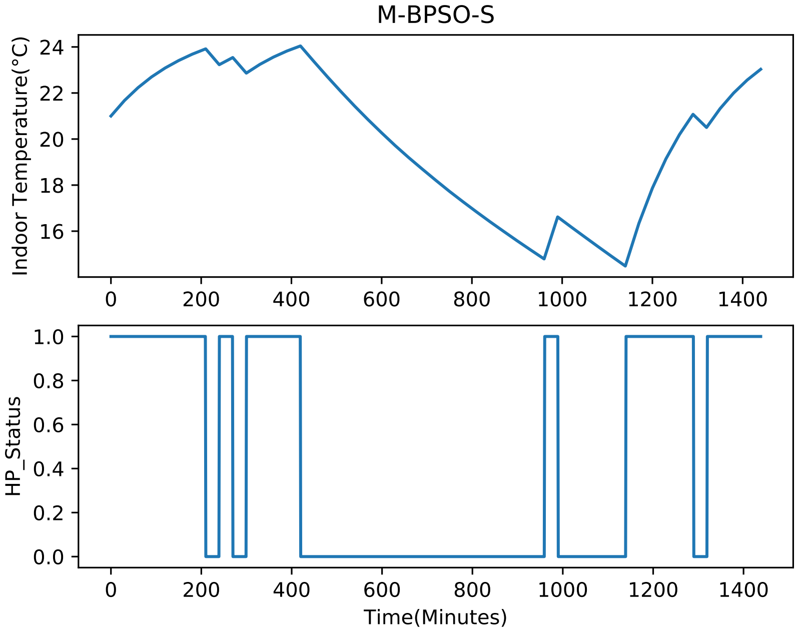


Figure 5.6: Heat pump schedule and corresponding indoor temperature (N/30 M-BPSO-S)

Results obtained by M-BPSO-S algorithm during the 50 runs are illustrated in Fig. 5.9 and Fig. 5.10. The graphs show that this algorithm suffers from inconsistency.

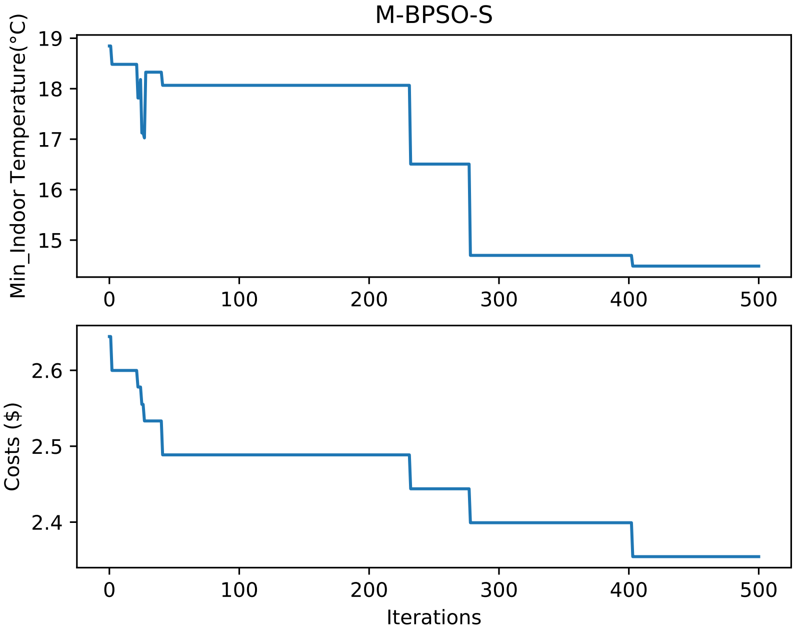


Figure 5.7: Minimum indoor temperature and cost variation (N/30 M-BPSO-S)

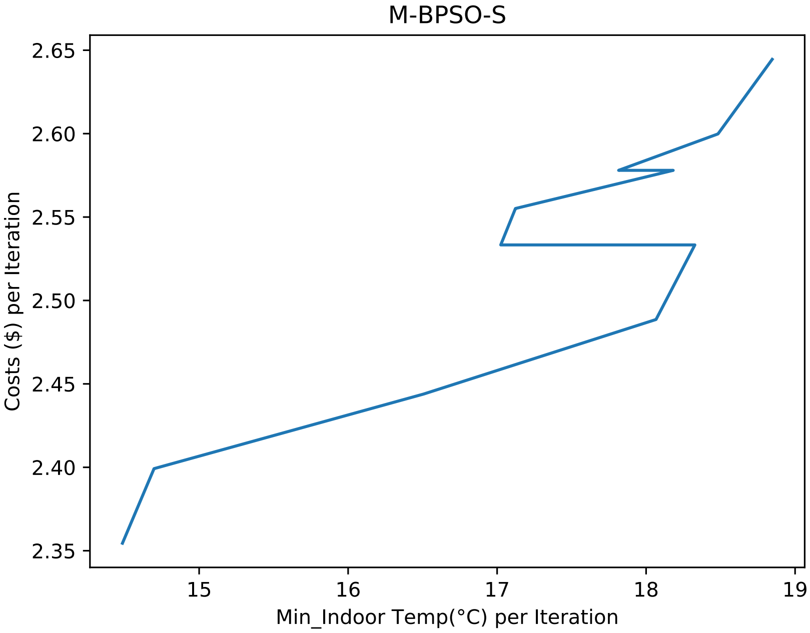
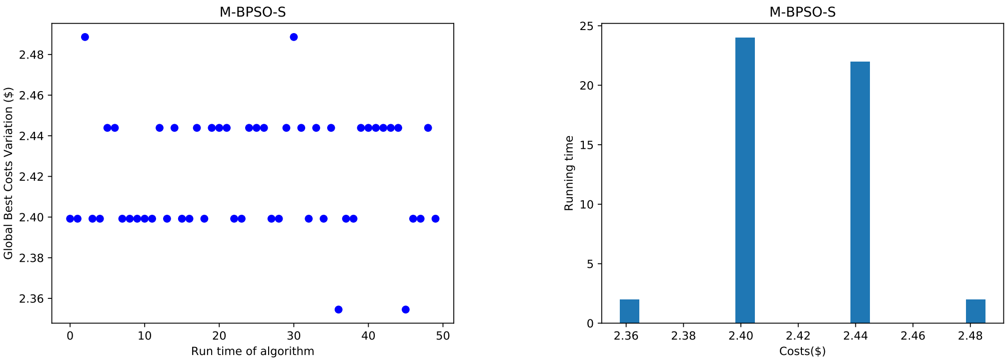


Figure 5.8: Search space trajectory of cost vs. minimum indoor temperature (N/30 M-BPSO-S)

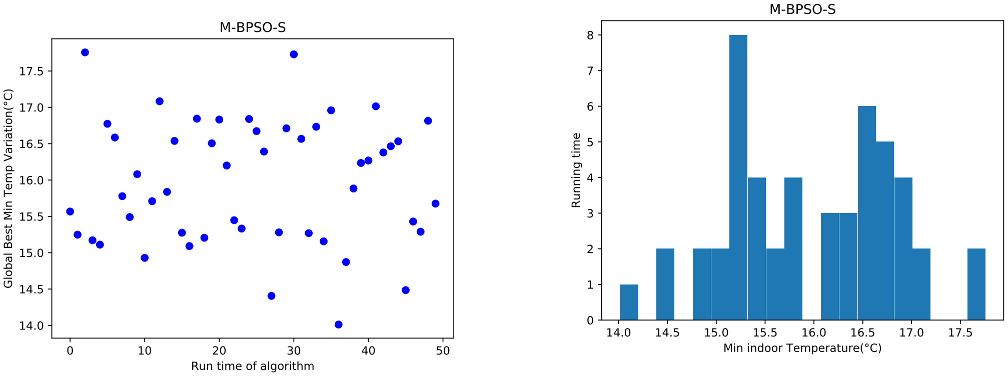
The histograms show that in most runs the algorithm converged costs around 2.4 $, while the minimum temperature range varies more widely from 14 °C to 17.5 °C.



(a) Costs

(b) Histogram of the costs

Figure 5.9: The best achieved costs during each run (N/30 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.10: The best achieved min temp during each run (N/30 M-BPSO-S)

5.1.1.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature profile corresponding to the best result achieved by M-BPSO-V are shown in Fig. 5.11. As the other algorithms, M-BPSO-V attempts to minimize cost, therefore the heat pump is kept continuously off for a long time, especially during the time with relatively higher electricity prices.

The variation of minimum indoor temperature and cost during the optimization progress is illustrated in Fig. 5.12. The M-BPSO-V algorithm has converged to the optimum cost and corresponding indoor temperature after 31 iterations. It is a slightly slower than GA, but significantly better than M-BPSO-S. More information about minimum indoor temperature changes versus electricity cost is provided in Fig. 5.13.

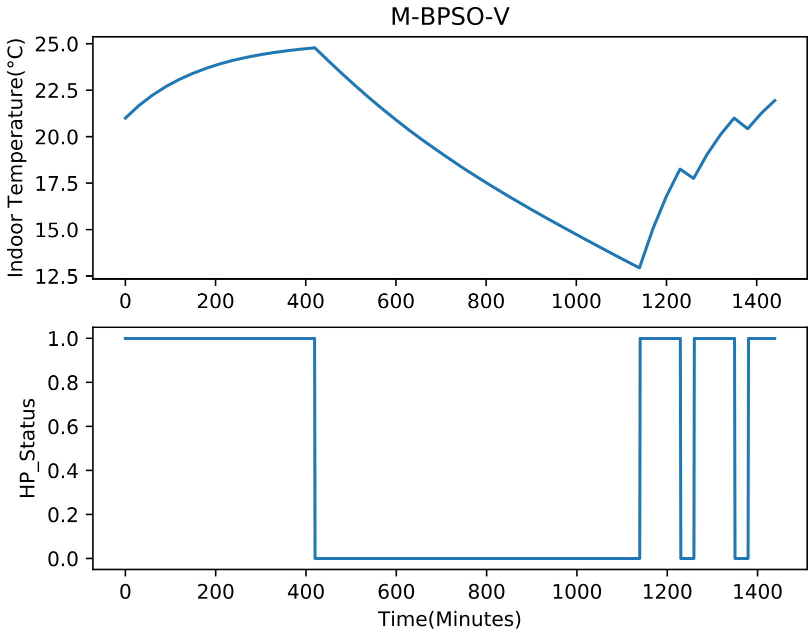


Figure 5.11: Heat pump schedule and corresponding indoor temperature (N/30 M-BPSO-V)

Fig. 5.14 and Fig. 5.15 show the costs and the minimum indoor temperature values achieved by M-BPSO-V over the 50 runs. Both graphs and histograms show high consistency of results provided by this algorithm.

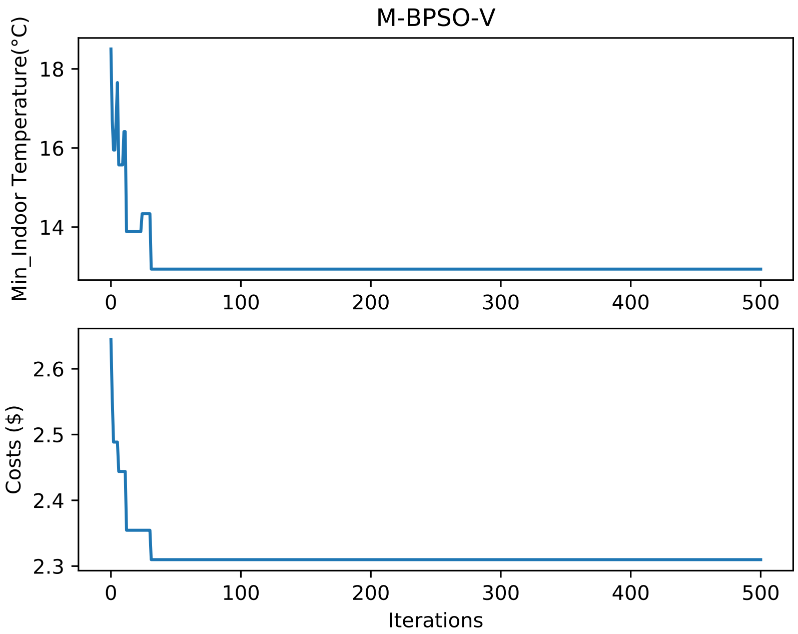


Figure 5.12: Minimum indoor temperature and cost variation (N/30 M-BPSO-V)

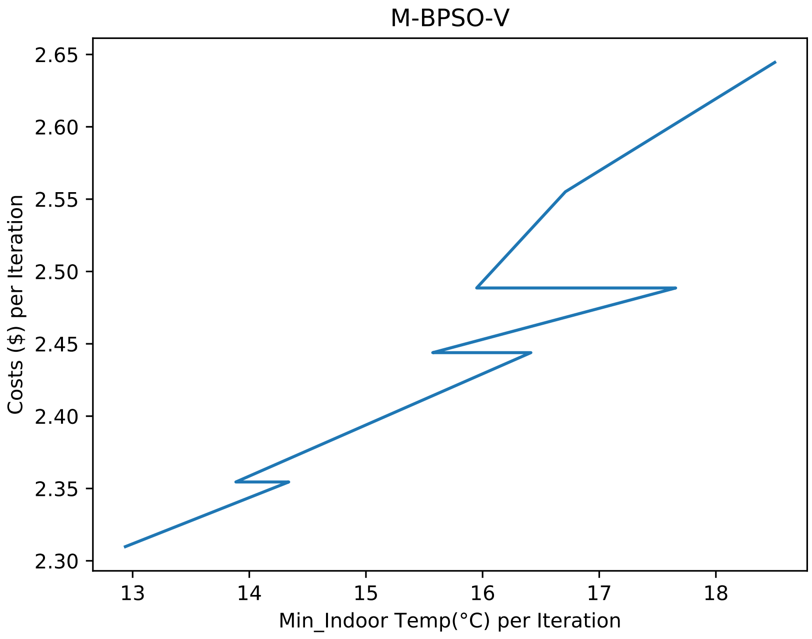
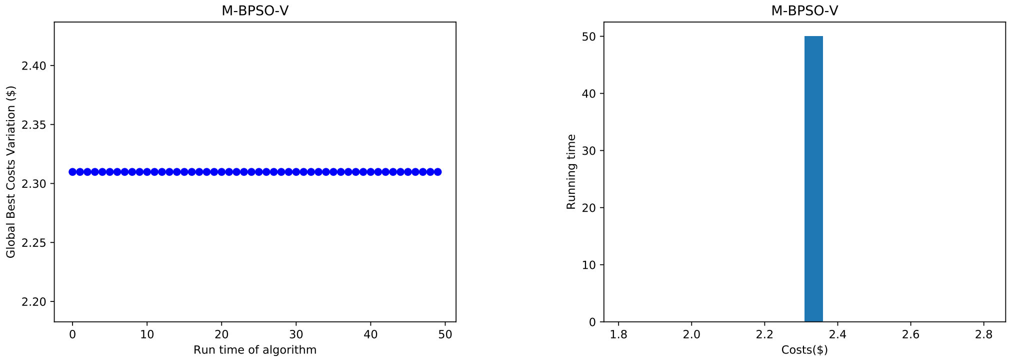


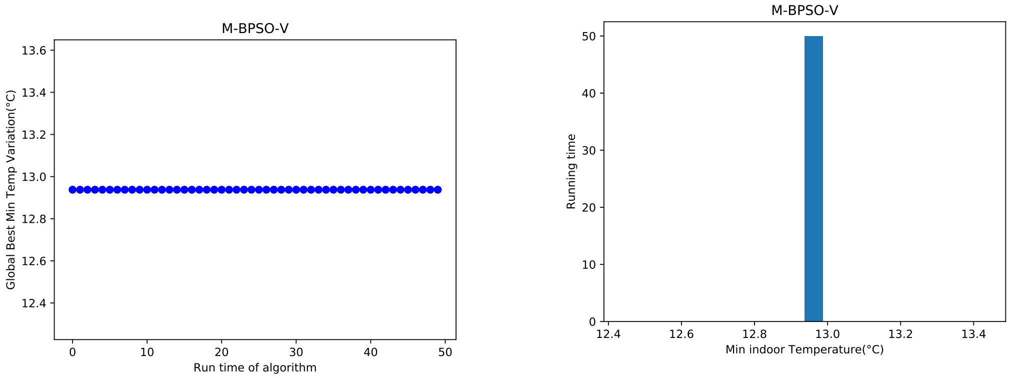
Figure 5.13: Search space trajectory of cost vs. minimum indoor temperature (N/30 M-BPSO-V)



(a) Costs

(b) Histogram of the costs

Figure 5.14: The best achieved costs during each run (N/30 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.15: The best achieved min temp during each run (N/30 M-BPSO-V)

5.1.2 Optimization of HP with 15 minutes time interval resolution

In this scenario the time interval resolution is set to 15 minutes. Accordingly, the minimum running time of the heat pump is 15 minutes.

As previously explained, for simulating the heat pump scheduling, the required running time of the heat pump during the optimization period should be determined considering the heating energy demand. Since in this scenario the time interval resolution has changed, all analyses regarding the heat loss flow and required heat pump running time must be updated. Based on (5.1) and (5.2), the updated total amount of heat loss during the optimization period is, $Q_{\mathrm{LS}}^{total} = 109.987MJ$. The heat flow of heat pump is the same as before, $Q_{\mathrm{HP}} = 10.383MJ/h$, because there are no changes in heat pump parameters and reference temperature setting. Given the total amount of heat loss and heat flow of the heat pump, the required heat pump running time is

$$\begin{aligned} HP_{\mathrm{RRT}} &= \frac{Q_{\mathrm{LS}}^{total}}{Q_{\mathrm{HP}}} \\ &= 10.59hrs, \end{aligned} \tag{5.4}$$

corresponding to 43 time slots based on 15 minutes time interval resolution and the the minimum running time constraint. While the amount of heat loss and the required running time of the heat pump is nearly the same as the prior scenario, the required number of running time-slots not doubled despite doubling the time resolution in comparison to the previous scenario. It is expected that the higher time interval resolution will lead to a more accurate mapping between running time and running time-slots and result in more energy cost savings.

The results of heat pump scheduling simulation using GA, M-BPSO-S and M-BPSO-V algorithms are presented in the following sections.

5.1.2.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.16. Since in this case, the maximum allowed continuous off-time constrain was not applied, the heat pump was kept continuously off for a long duration of time, mainly during the higher electricity price intervals.

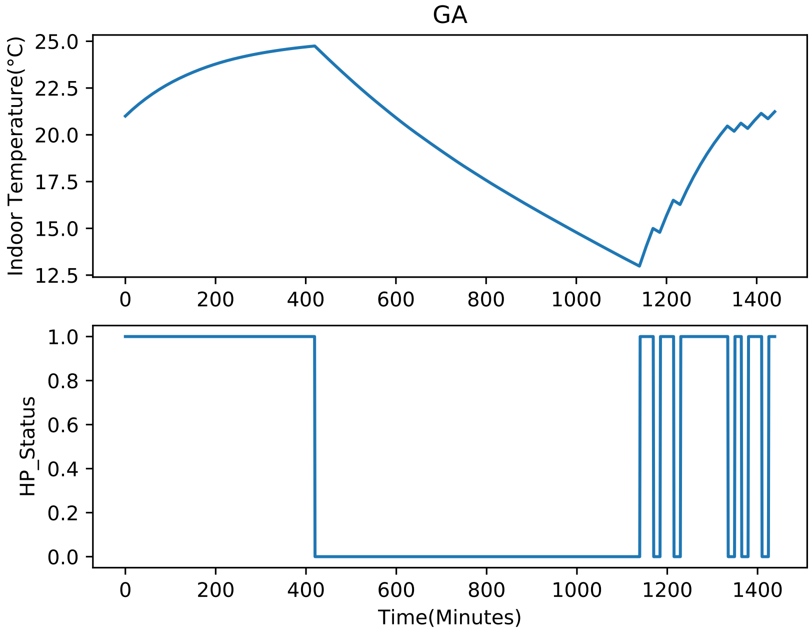


Figure 5.16: Heat pump schedule and corresponding indoor temperature (N/15 GA)

Fig. 5.17 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. Based on the graph, the GA converged to the optimum cost and corresponding indoor temperature in less than 100 iterations.

More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.18.

As mentioned before, due to the stochastic nature of metaheuristic algorithm, the HP scheduling simulation using GA was executed 50 times and the best results are presented. The performance of GA in terms of best obtained cost and minimum

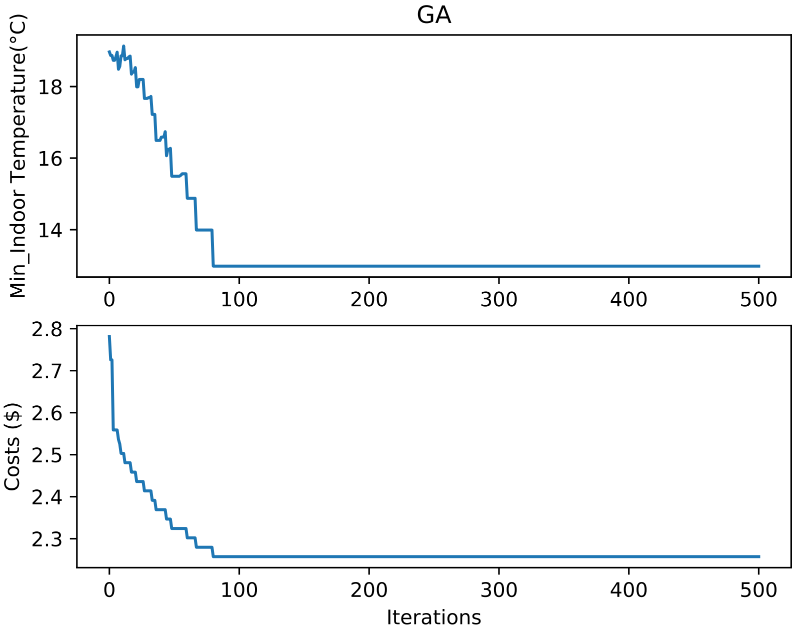


Figure 5.17: Minimum indoor temperature and cost variation (N/15 GA)

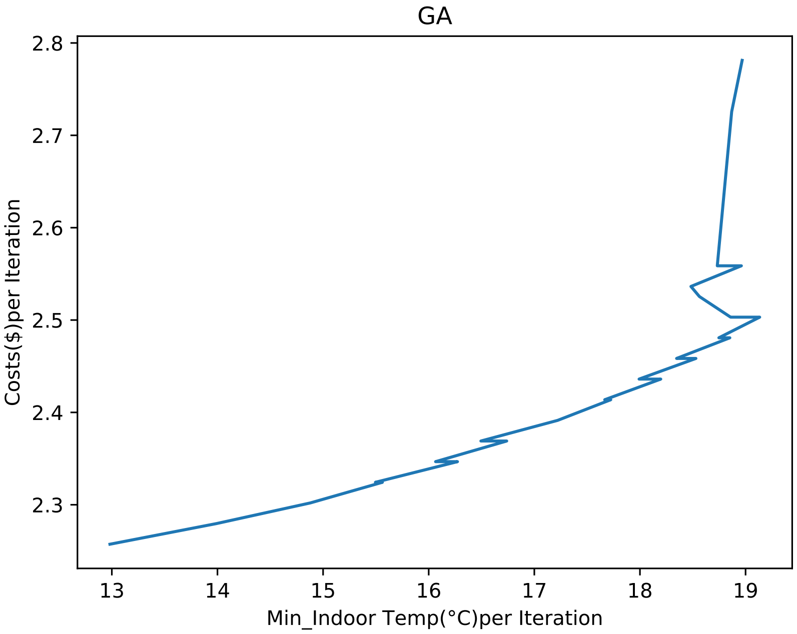
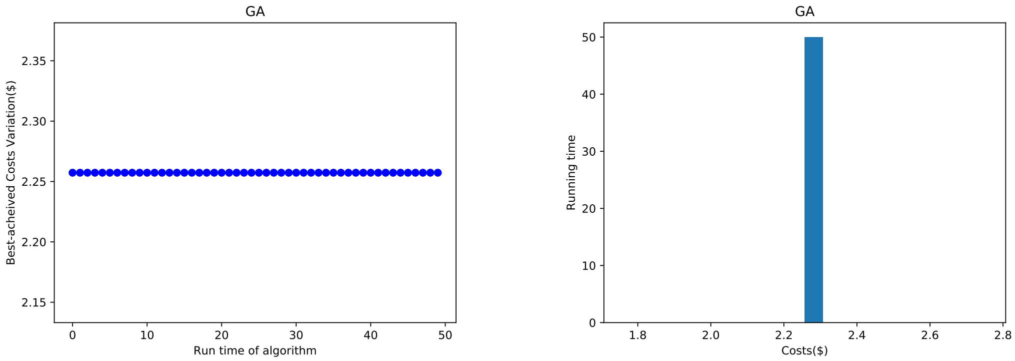


Figure 5.18: Search space trajectory of cost vs. minimum indoor temperature (N/15 GA)

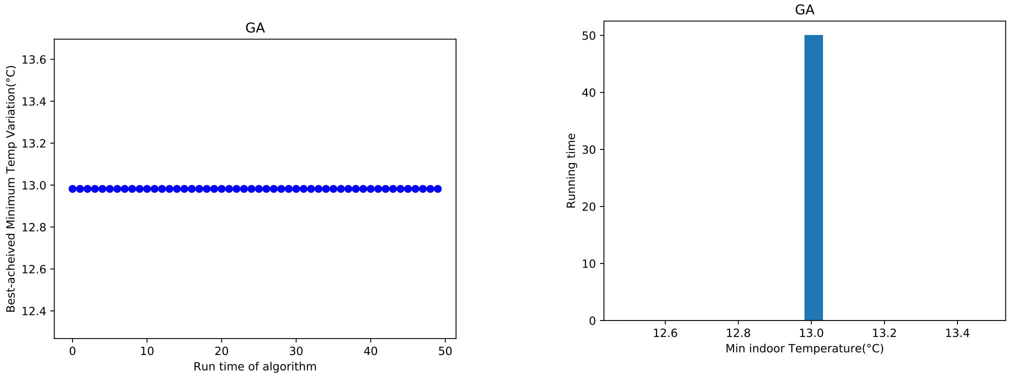
indoor temperature over the course of 50 runs are shown in Fig. 5.19 and Fig. 5.20. The graphs and histograms show that the algorithm successfully converged to the global optimum cost and minimum indoor temperature in all runs.



(a) Costs

(b) Histogram of the costs

Figure 5.19: The best achieved costs during each run (N/15 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.20: The best achieved min temp during each run (N/15 GA)

5.1.2.2 Results using M-BPSO-S

The heat pump running schedule and indoor temperature corresponding to the best result found by M-BPSO-S are shown in Fig. 5.21. Since the algorithm looks to minimize the cost, the heat pump is kept continuously off during several periods. The algorithm did not converge to the optimum point.

The variation of minimum indoor temperature and costs over 500 iterations are shown in Fig. 5.22. It is clear that after 100 iteration the algorithm got trapped in local optima and was not able to escape during the remaining simulation time.

The variation of minimum indoor temperature versus electricity cost during the optimization progress is shown in Fig. 5.23. The graph shows that cost reduction leads to indoor temperature decrease which has negative effect on the thermal comfort of occupants.

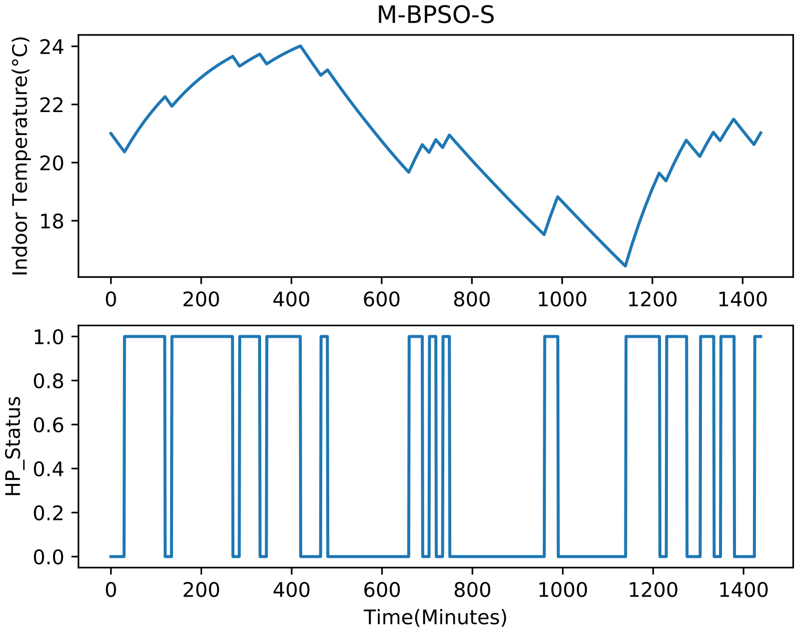


Figure 5.21: Heat pump schedule and corresponding indoor temperature (N/15 M-BPSO-S)

More details regarding the obtained results by M-BPSO-S algorithm provided in Fig. 5.24 and Fig. 5.25. The graphs and histograms show that algorithm was not

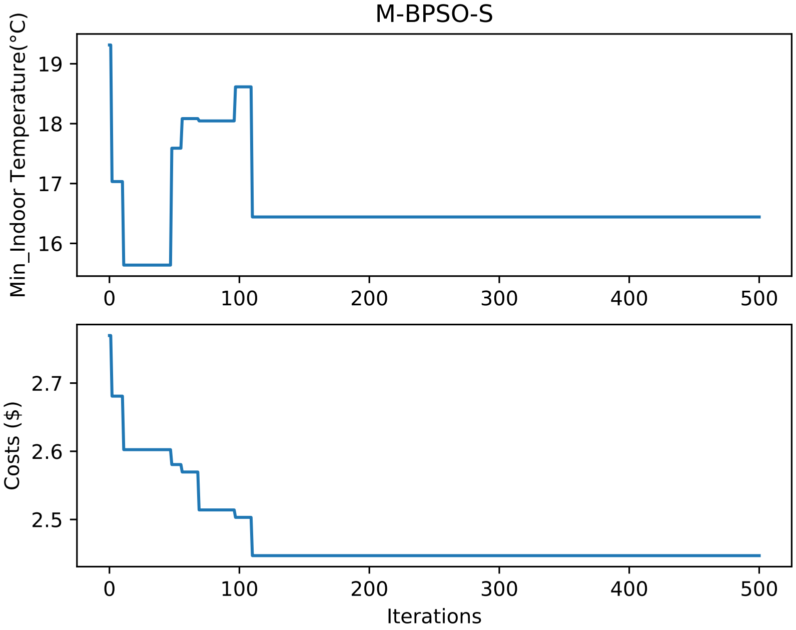


Figure 5.22: Minimum indoor temperature and cost variation (N/15 M-BPSO-S)

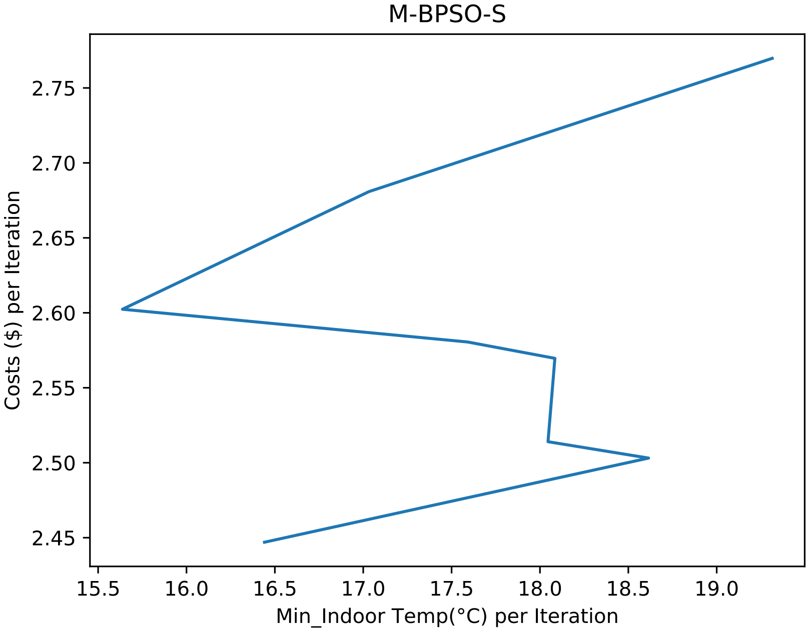
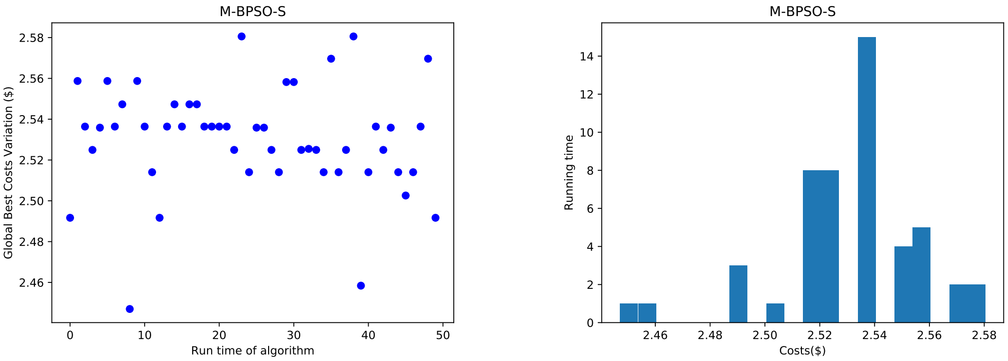


Figure 5.23: Search space trajectory of cost vs. minimum indoor temperature (N/15 M-BPSO-S)

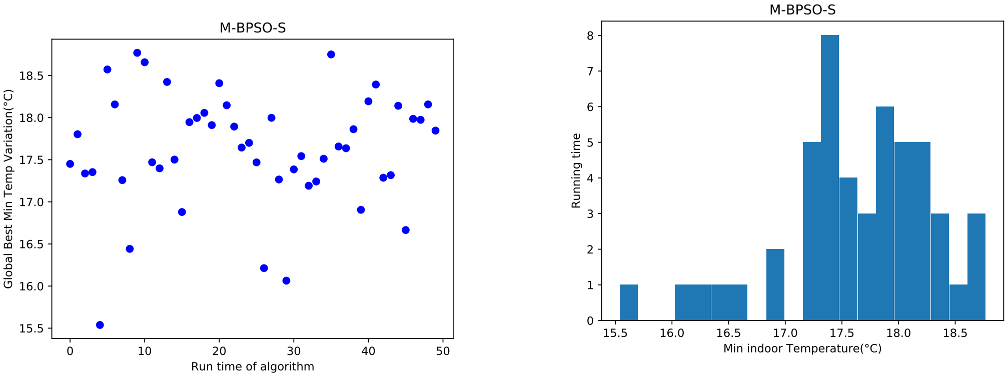
consistent and got trapped in different local optima points during the optimization progress. In more than 60% of runs, the algorithm converged to cost around 2.53 $, which is far from the optimum point.



(a) Costs

(b) Histogram of the costs

Figure 5.24: The best achieved costs during each run (N/15 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.25: The best achieved min temp during each run (N/15 M-BPSO-S)

5.1.2.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature corresponding to the best result found by M-BPSO-V are shown in Fig. 5.26. The optimized schedule is nearly the same as the schedule found using the GA, keeping the heat pump continuously off for a long period to minimize the overall energy cost.

The variation of minimum indoor temperature and cost during the optimization progress is illustrated in Fig. 5.27. The M-BPSO-V algorithm has converged to the optimum cost after about 100 iterations, slower than the GA. More information about minimum indoor temperature changes versus electricity cost, is provided in Fig. 5.28.

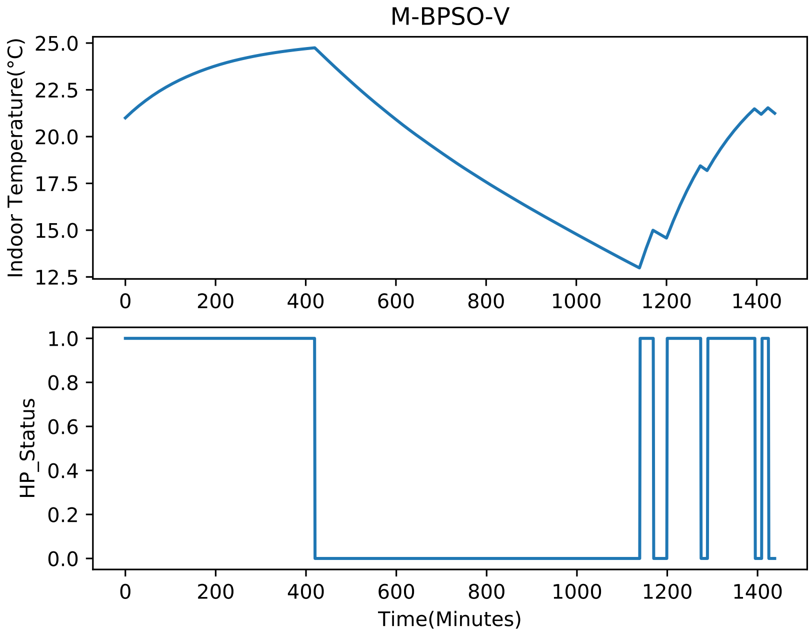


Figure 5.26: Heat pump schedule and corresponding indoor temperature (N/15 M-BPSO-V)

As the other algorithms, the M-BPSO-V algorithm was executed for 50 times. More information on the distribution of the results during this set of simulations is provided in Fig. 5.29 and Fig. 5.30. The graphs and histograms illustrate that the algorithm was fully consistent since it converged to the global optimum in all runs.

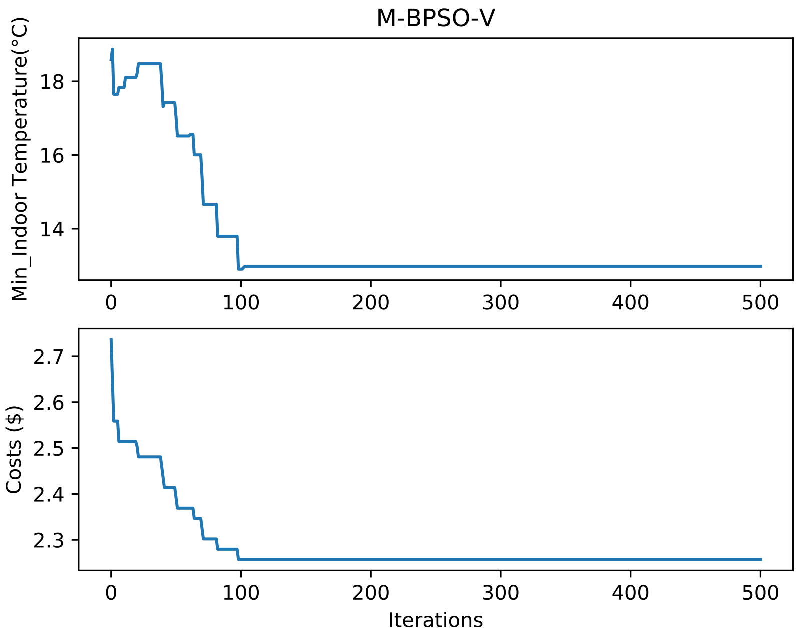


Figure 5.27: Minimum indoor temperature and cost variation (N/15 M-BPSO-V)

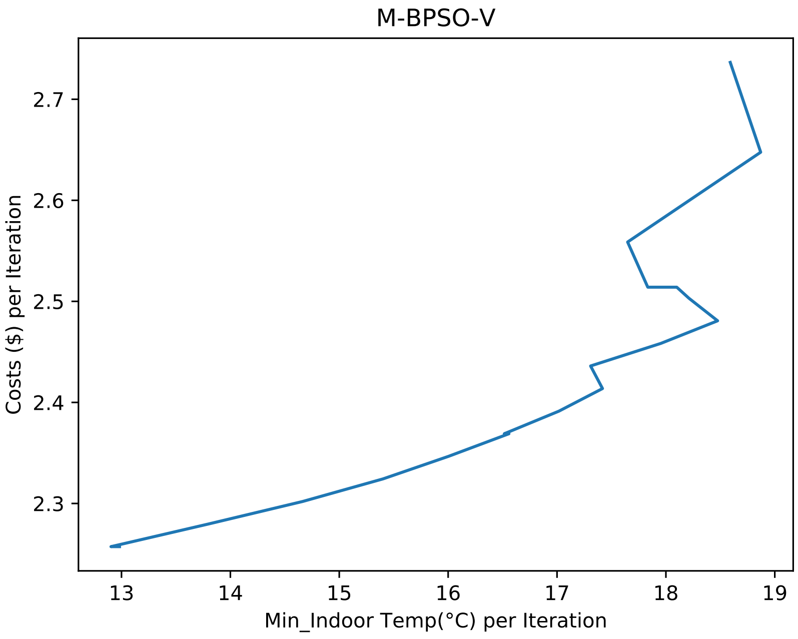
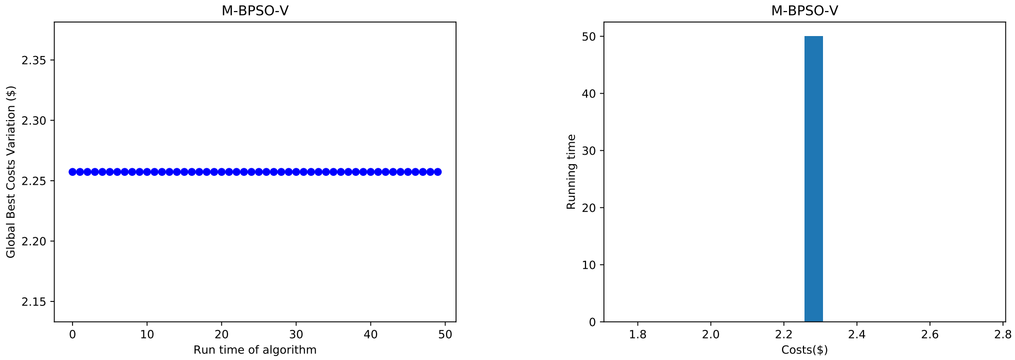


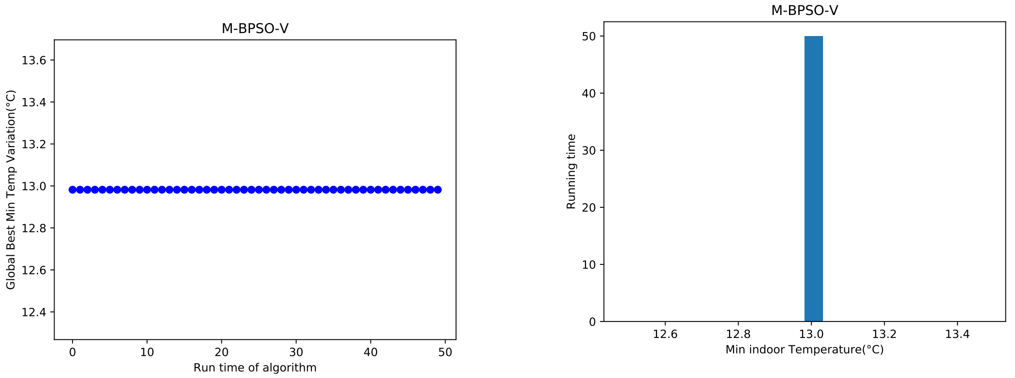
Figure 5.28: Search space trajectory of cost vs. minimum indoor temperature (N/15 M-BPSO-V)



(a) Costs

(b) Histogram of the costs

Figure 5.29: The best achieved costs during each run (N/15 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.30: The best achieved min temp during each run (N/15 M-BPSO-V)

5.2 Optimization of HP Schedule Considering Occupant Comfort with Static off-time

In this scenario, the thermal comfort of occupant is considered through the application of the fixed maximum allowed continuous off-time of heat pump constraint.

5.2.1 Optimization of HP with 30 minutes time interval resolution (C/S/30)

In this scenario the time slot resolution as thus minimum running time of the heat pump is set to 30 minutes.

As previously explained, for simulating the heat pump scheduling, the required running time of the heat pump during the optimization period must be determined considering the heating energy demand. Since the thermal dynamics of the building in this scenario is the same as for N/30 scenario, the analyses regarding the heat loss flow and required heat pump running time are not updated and the same values are used. The difference between this scenario and N/30 scenario is the consideration of occupant comfort based on static off-time approach. Hence, the maximum allowed continuous off-time of heat pump with respect to the reference temperature set point and the minimum acceptable temperature is set as 3 time slots. Details of the calculation of maximum continuous off-time of heat pump are presented in chapter 4.

5.2.1.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.31. Since in this case, the maximum allowed continuous off-time constrain was applied, the heat pump was not kept continuously off for a long period of time.

Fig. 5.32 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. As seen in the graph, the GA quickly converged to the

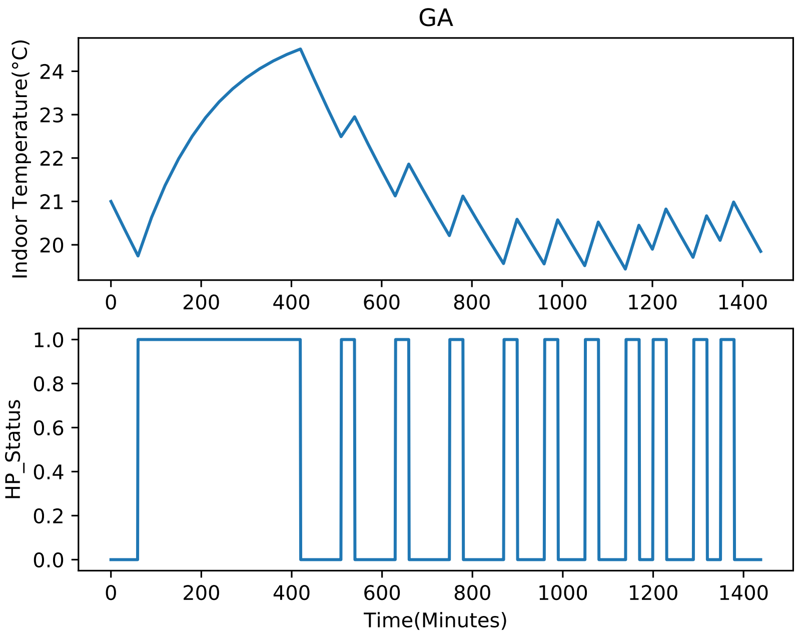


Figure 5.31: Heat pump schedule and corresponding indoor temperature (C/S/30 GA)

optimum cost and then it significantly improved the minimum indoor temperature over few iterations.

More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.33. It shows that, after converging to the cost optimum, the GA significantly improved minimum indoor temperature during the remaining iterations.

As mentioned before, the HP scheduling simulation using GA was executed 50 times and the best results are reported. The performance of GA in terms of best obtained cost and minimum indoor temperature over the 50 running times are illustrated in Fig. 5.34 and Fig. 5.35, respectively. The histograms show that in more than 25 runs, the GA was not able to escape the local optima.

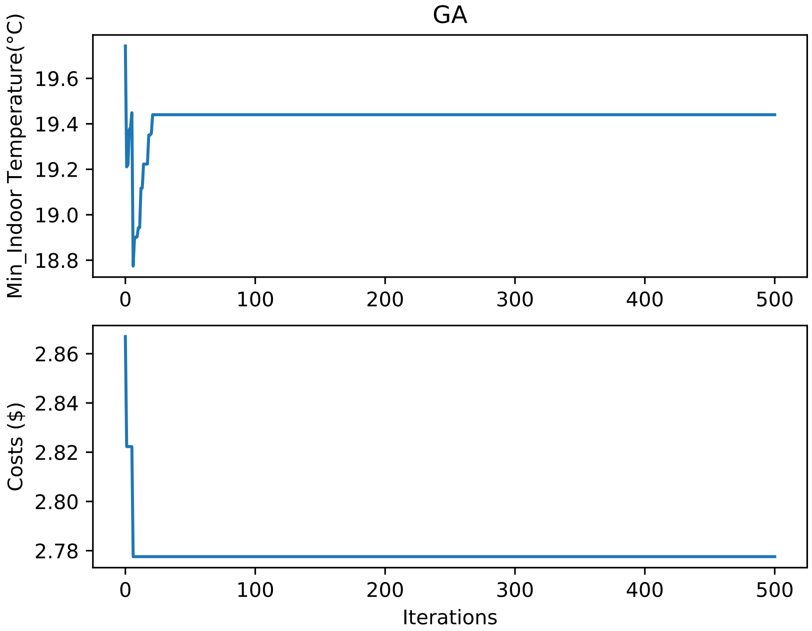


Figure 5.32: Minimum indoor temperature and cost variation (C/S/30 GA)

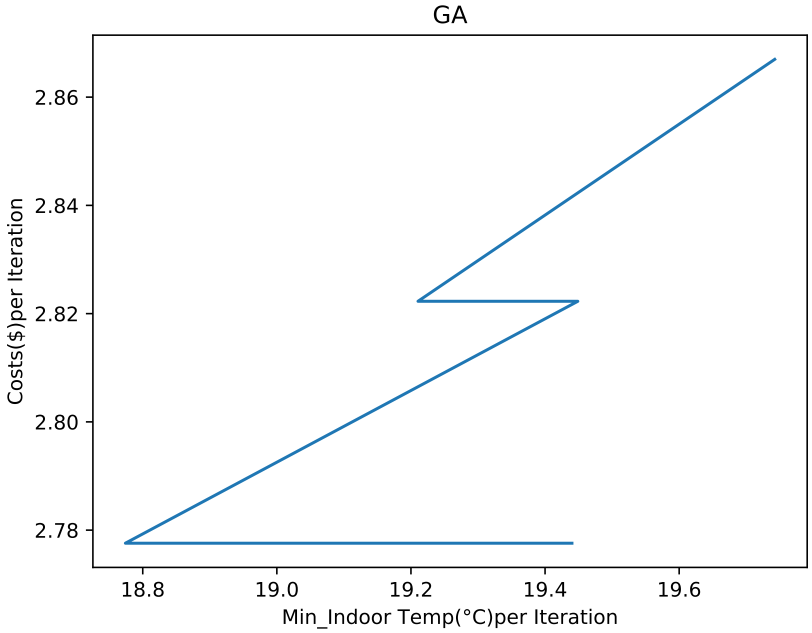
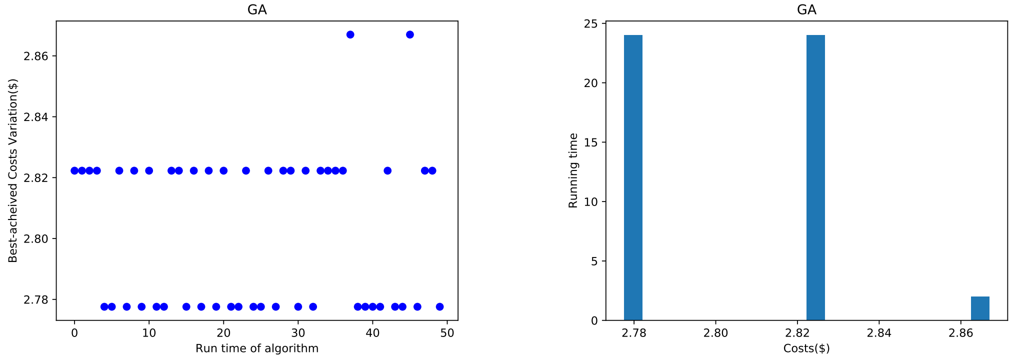


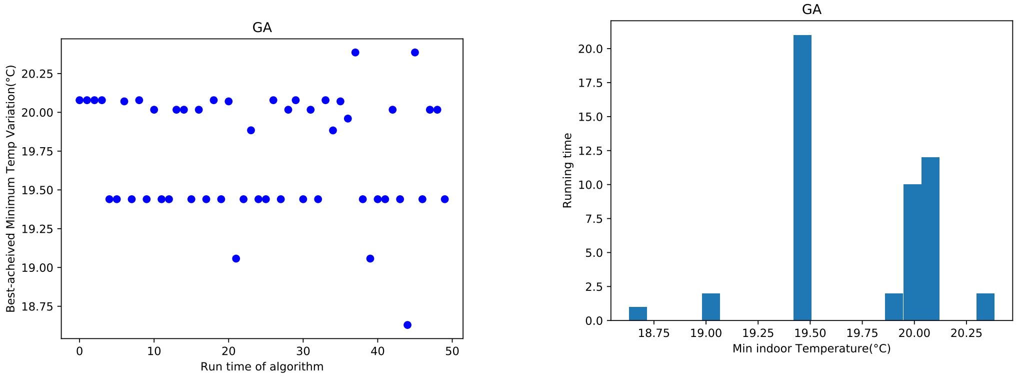
Figure 5.33: Search space trajectory of cost vs. minimum indoor temperature (C/S/30 GA)



(a) Costs

(b) Histogram of the costs

Figure 5.34: The best achieved costs during each run (C/S/30 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.35: The best achieved min temp during each run (C/S/30 GA)

5.2.1.2 Results using M-BPSO-S

Fig. 5.36 shows the heat pump running schedule and indoor temperature corresponding to the best result obtained using M-BPSO-S. The occupant thermal comfort is considered by enforcing the maximum continuous off state of the heat pump.

Fig. 5.37 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.38. The graph shows that the algorithm converged to the optimum cost after about 450 iterations but it could not enhance the temperature to reach to optimum value during the remain simulation time.

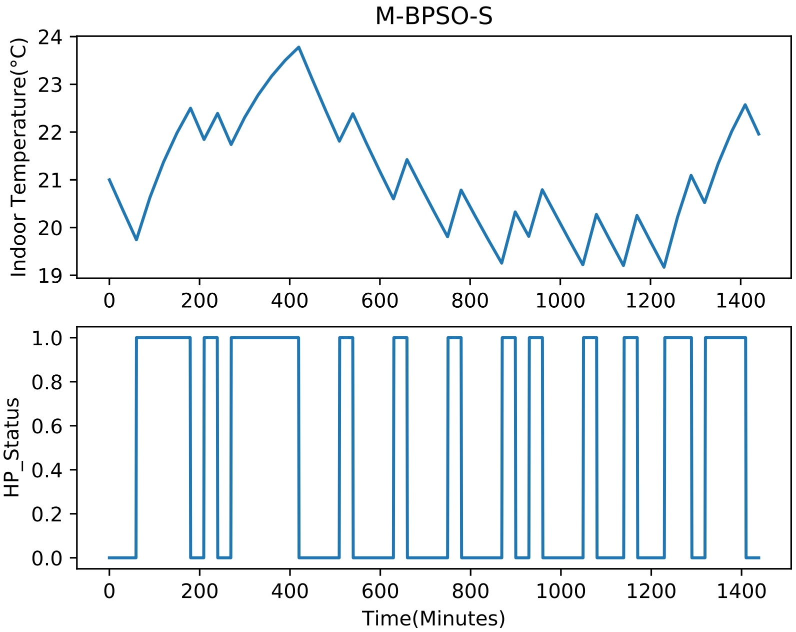


Figure 5.36: Heat pump schedule and corresponding indoor temperature (C/S/30 M-BPSO-S)

Since the scheduling simulation using M-BPSO-S was executed 50 times and the best records are presented, more information regarding the obtained results during whole execution period are presented in Fig. 5.39 and 5.40. The histograms show that in about 15 runs the algorithm reached to optimum costs but never converged to corresponding optimum temperature.

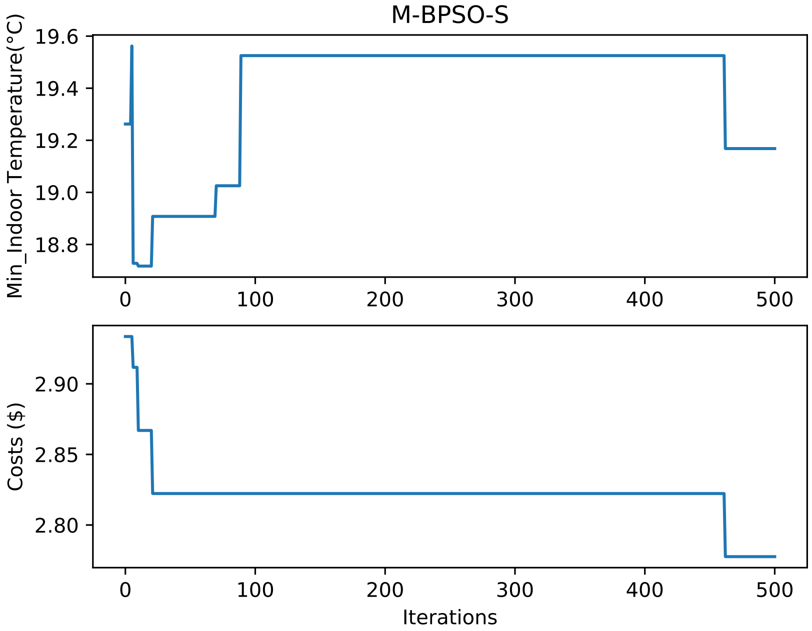


Figure 5.37: Minimum indoor temperature and cost variation (C/S/30 M-BPSO-S)

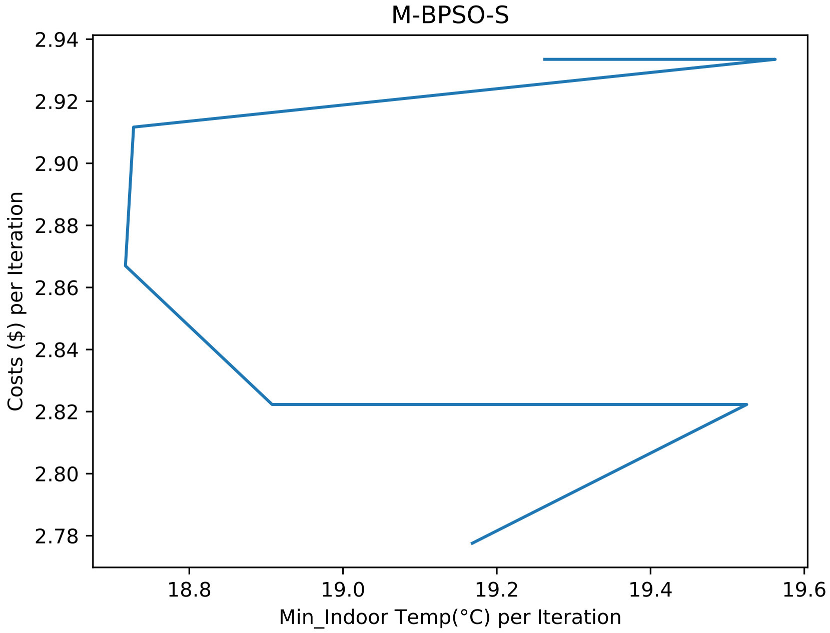
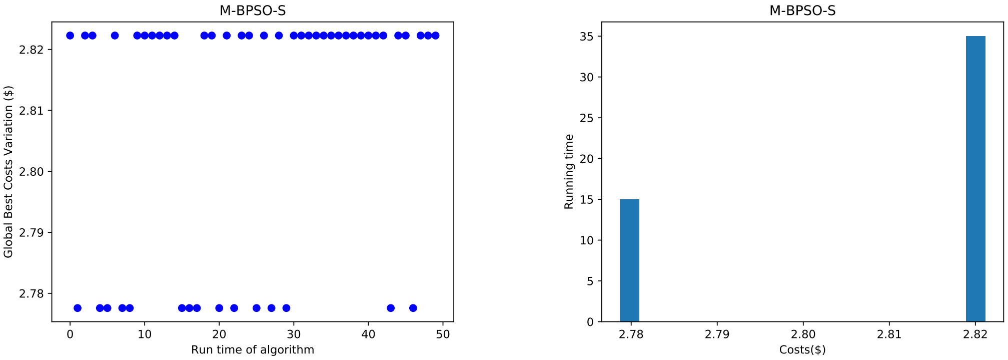


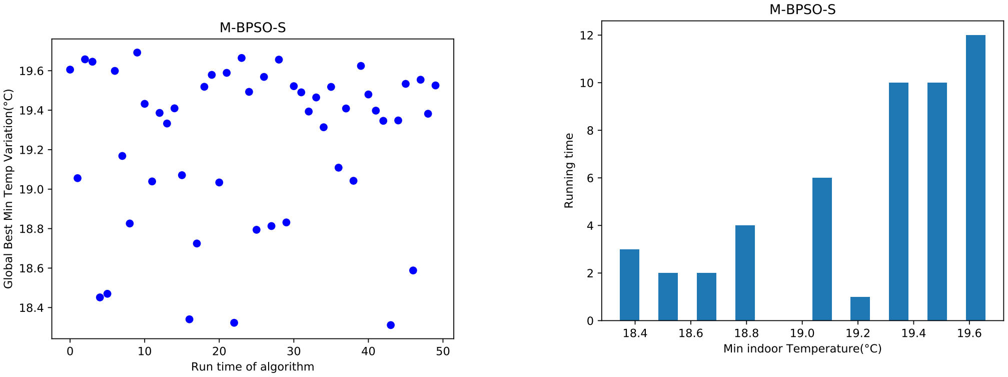
Figure 5.38: Search space trajectory of cost vs. minimum indoor temperature (C/S/30 M-BPSO-S)



(a) Costs

(b) Histogram of the costs

Figure 5.39: The best achieved costs during each run (C/S/30 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.40: C/S/30 M-BPSO-S-The best achieved min temp during each run (C/S/30 M-BPSO-S)

5.2.1.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the M-BPSO-V are illustrated in Fig. 5.41. Since in this case, the maximum allowed continuous off-time constrain was applied, the heat pump was not kept off for a long duration of time.

Fig. 5.42 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. While the cost optimization leads to temperature decrease, the algorithm successfully increases the minimum indoor temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.43. It shows that, after converging to the cost optimum, the M-BPSO-V significantly improved minimum indoor temperature during the remaining iterations.

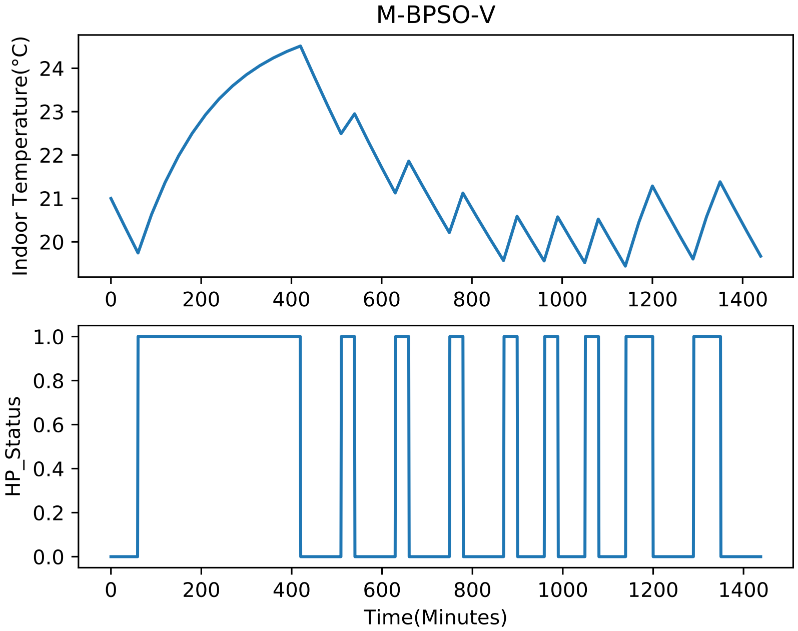


Figure 5.41: Heat pump schedule and corresponding indoor temperature (C/S/30 M-BPSO-V)

As mentioned before, the HP scheduling simulation using M-BPSO-V was executed 50 times and the best results are reported. The performance of M-BPSO-V in terms of best obtained cost and minimum indoor temperature over the 50 runs are shown

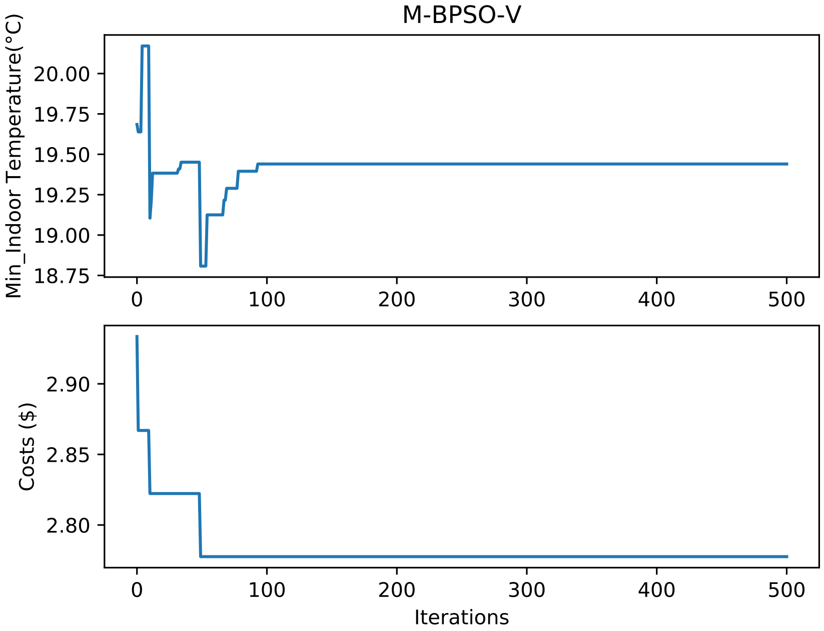


Figure 5.42: Minimum indoor temperature and cost variation (C/S/30 M-BPSO-V)

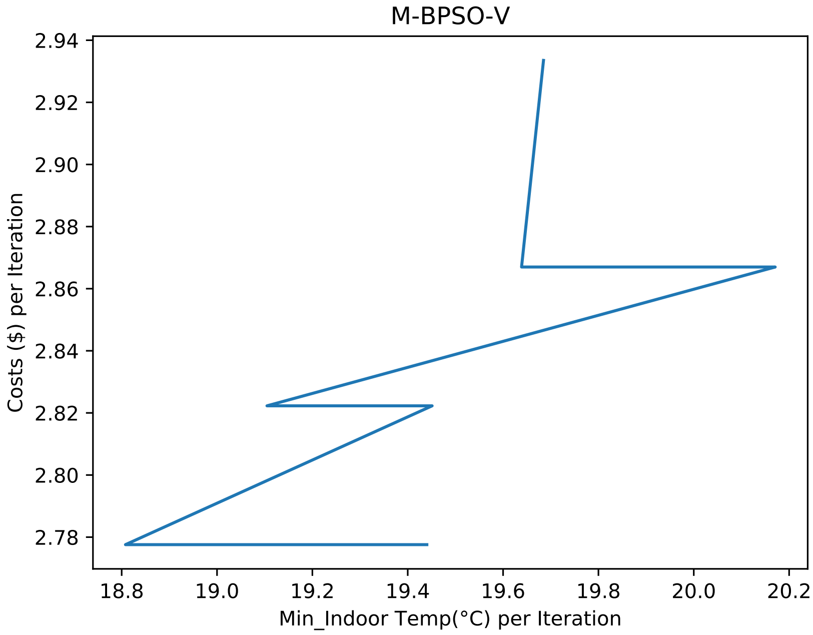
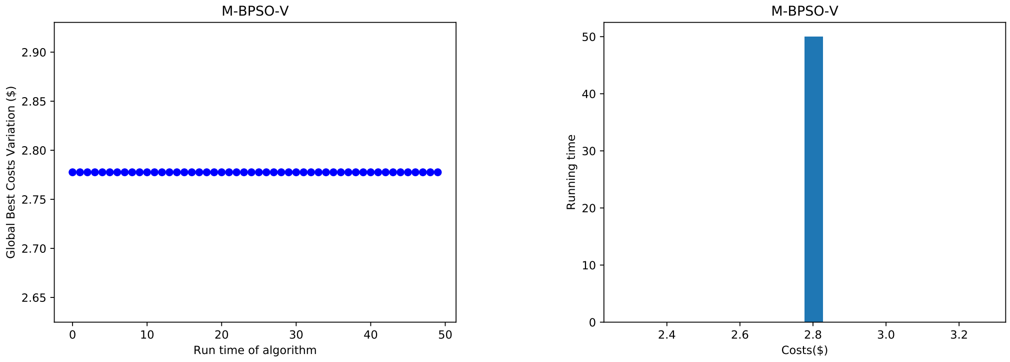


Figure 5.43: Search space trajectory of cost vs. minimum indoor temperature (C/S/30 M-BPSO-V)

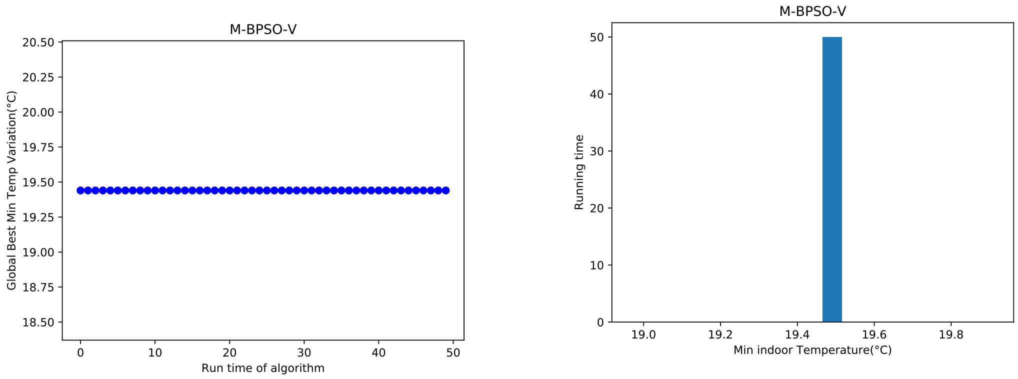
in Fig. 5.44 and Fig. 5.45, respectively. The histograms show that M-BPSO-V performed very well since it converged to global optimum in all runs.



(a) Costs

(b) Histogram of the costs

Figure 5.44: The best achieved costs during each run (C/S/30 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.45: The best achieved min temp during each run (C/S/30 M-BPSO-V)

5.2.2 Optimization of HP with 15 minutes time interval resolution (C/S/15)

In this scenario, the time interval resolution and thus the minimum running time of the heat pump is set to 15 minutes.

As mentioned earlier, for simulating the heat pump scheduling, the required running time of the heat pump during the optimization period must be determined considering the heating energy demand. Since the thermal dynamics of the building in this scenario is the same as for N/15 scenario, the analyses regarding the heat loss flow and required heat pump running time are not updated and the same values are used. The difference between this scenario and N/15 scenario is the consideration of occupant comfort based on static off-time approach. Hence, the maximum allowed continuous off-time of heat pump with respect to the reference temperature set point and the minimum acceptable temperature is set as 6 time slots. Details of the calculation of maximum continuous off-time of heat pump are presented in chapter 4.

5.2.2.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.46. Since in this case, the maximum allowed continuous off-time constrain was applied, the heat pump was not kept off for a long period of time.

Fig. 5.47 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. As seen in the graph, the GA reduced the cost and tried to improve the minimum temperature during the remain simulation time. While, GA slightly improved the minimum indoor temperature for obtained cost, it was not able to find global optima in terms of cost and corresponding minimum temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.48.

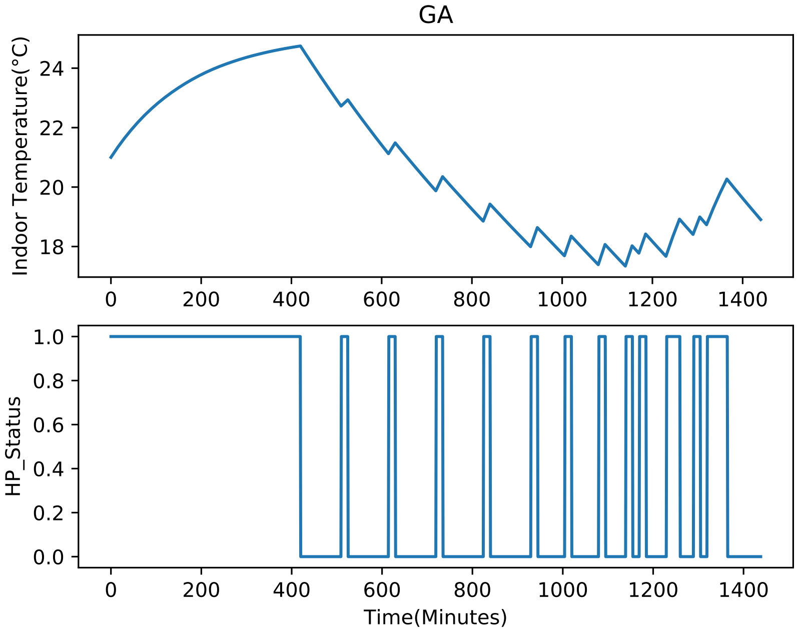


Figure 5.46: Heat pump schedule and corresponding indoor temperature (C/S/15 GA)

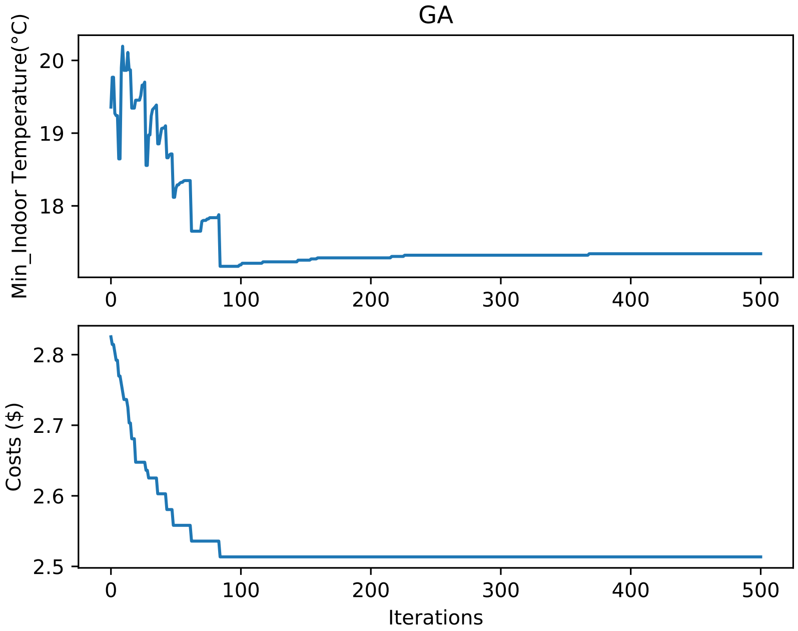


Figure 5.47: GA-Minimum indoor temperature and cost variation (C/S/15 GA)

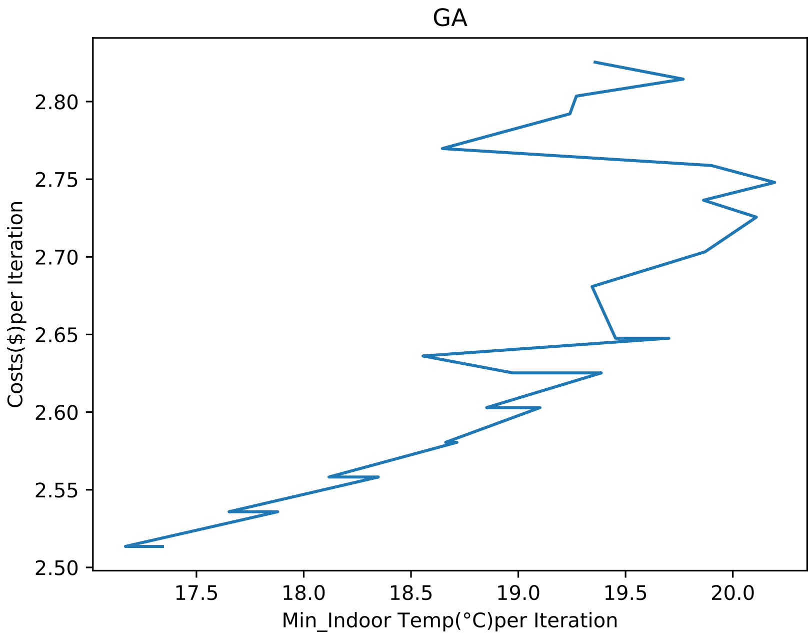
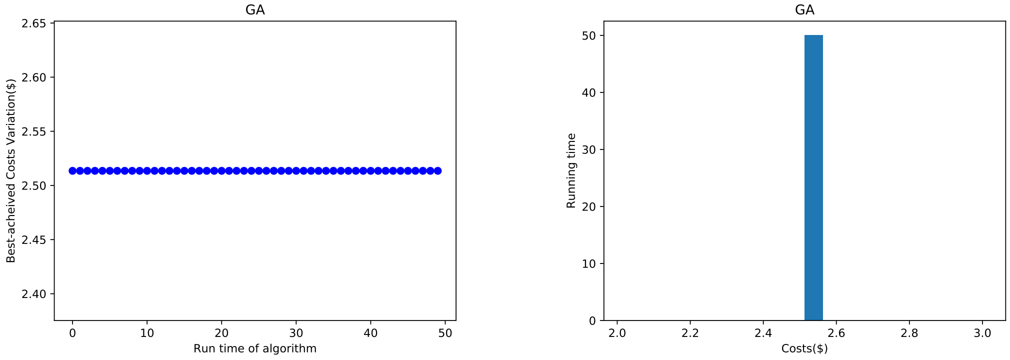


Figure 5.48: Search space trajectory of cost vs. minimum indoor temperature (C/S/15 GA)

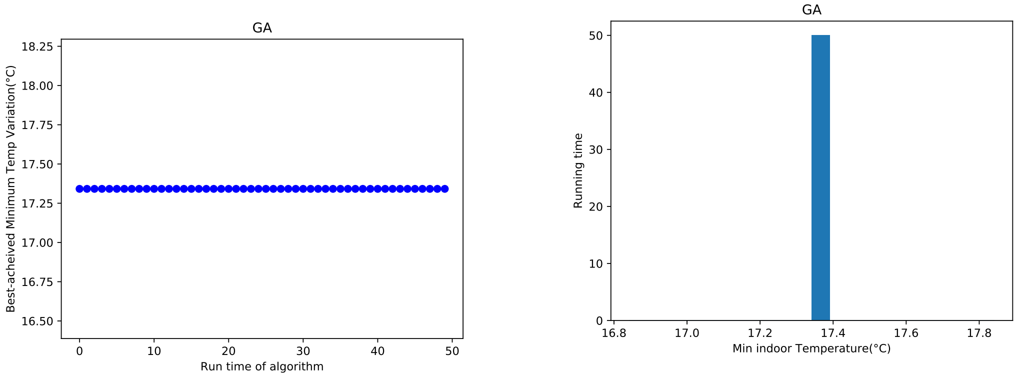
As mentioned earlier, the HP scheduling simulation using GA was executed 50 times and the best results reported. The performance of GA in terms of best obtained cost and minimum indoor temperature over the 50 running times are illustrated in 5.49 and Fig. 5.50, respectively. The histograms show that in all runs, GA was not able to reach global optima and converged to a close sub-optimal point.



(a) Costs

(b) Histogram of the costs

Figure 5.49: The best achieved costs during each run (C/S/15 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.50: The best achieved min temp during each run (C/S/15 GA)

5.2.2.2 Results using M-BPSO-S

Fig. 5.51 shows the heat pump running schedule and indoor temperature corresponding to the best result obtained using M-BPSO-S. The occupant thermal comfort is considered by enforcing the maximum continuous off state of the heat pump.

Fig. 5.52 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. The graph shows that algorithm was not able to converge to optimum cost and corresponding minimum indoor temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.53.

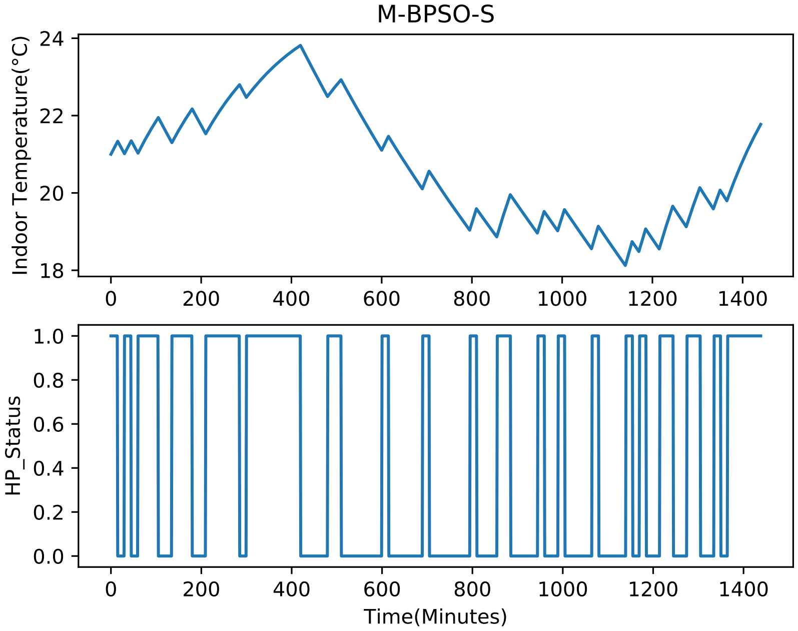


Figure 5.51: Heat pump schedule and corresponding indoor temperature (C/S/15 M-BPSO-S)

Since the scheduling simulation using M-BPSO-S was executed 50 times and the best records reported, more information regarding the obtained results during the whole execution period are presented in Fig. 5.54 and Fig. 5.55. The Graphs show that algorithm was not consistent and got trapped in local optima during each run.

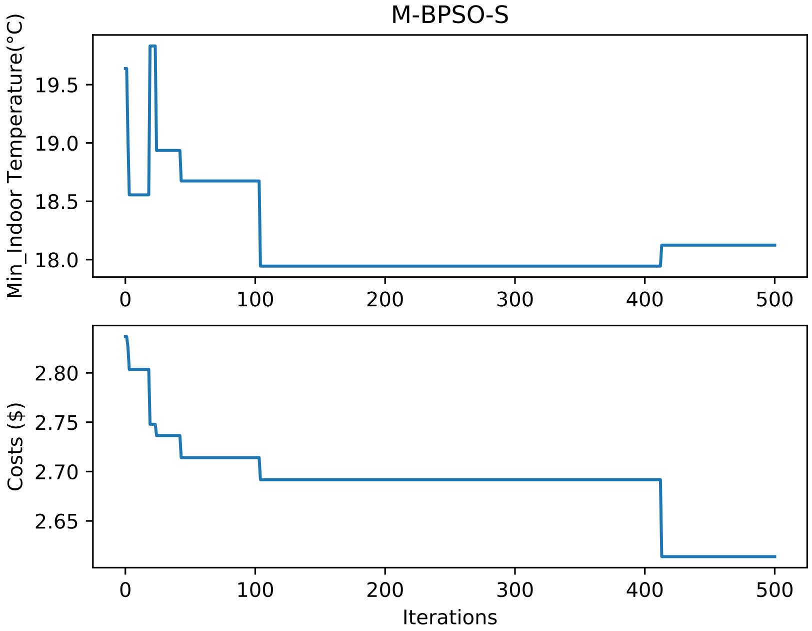


Figure 5.52: Minimum indoor temperature and cost variation (C/S/15 M-BPSO-S)

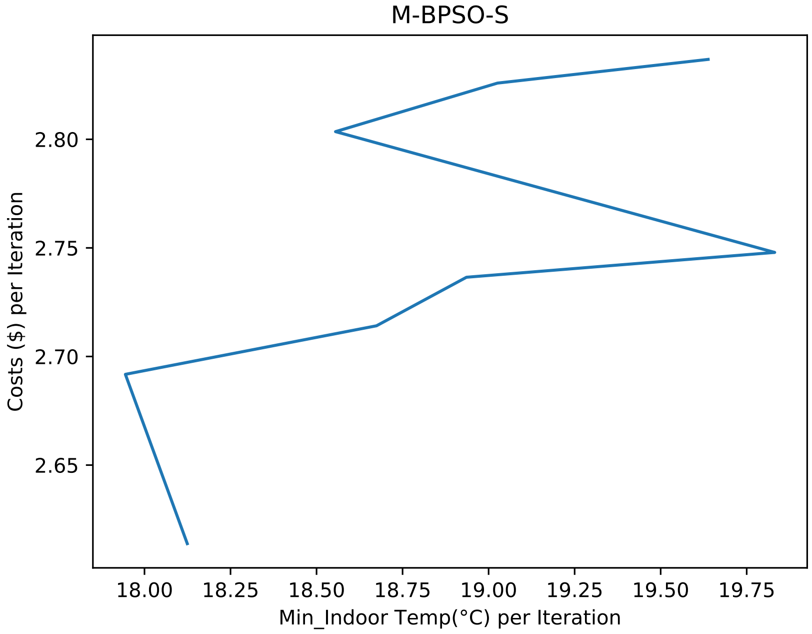
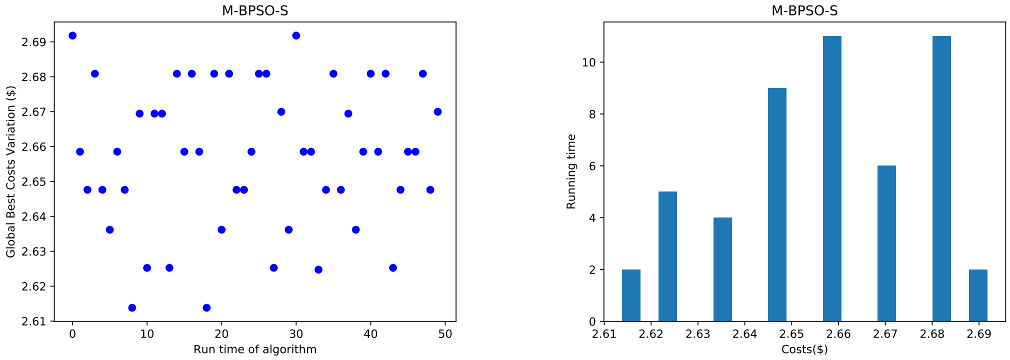


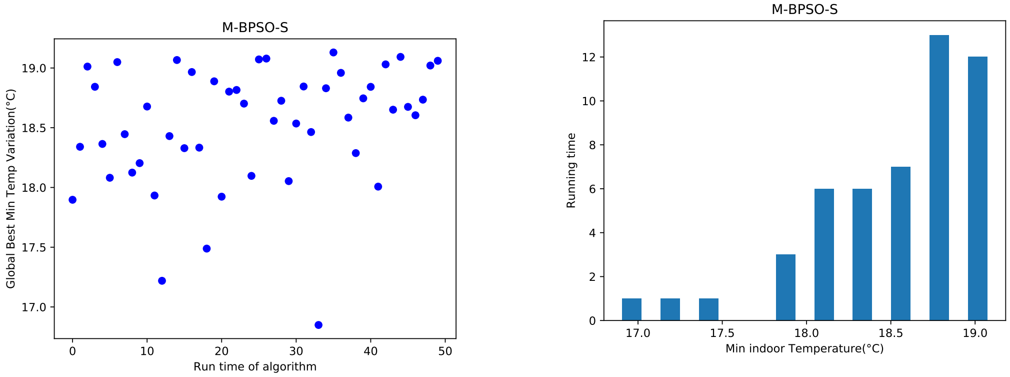
Figure 5.53: Search space trajectory of cost vs. minimum indoor temperature (C/S/15 M-BPSO-S)



(a) Costs

(b) Histogram of the costs

Figure 5.54: The best achieved costs during each run (C/S/15 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.55: The best achieved min temp during each run (C/S/15 M-BPSO-S)

5.2.2.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the M-BPSO-V are illustrated in Fig. 5.56. Since in this scenario, the maximum allowed continuous off-time constrain was applied, the heat pump was not kept off for a long duration of time.

Fig. 5.57 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. Based on the graph, the M-BPSO-V converged to the optimum cost and corresponding temperature after about 250 iterations.

More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.58. The graph shows that the algorithm attempts to improve minimum indoor temperature while minimizing the operation cost.

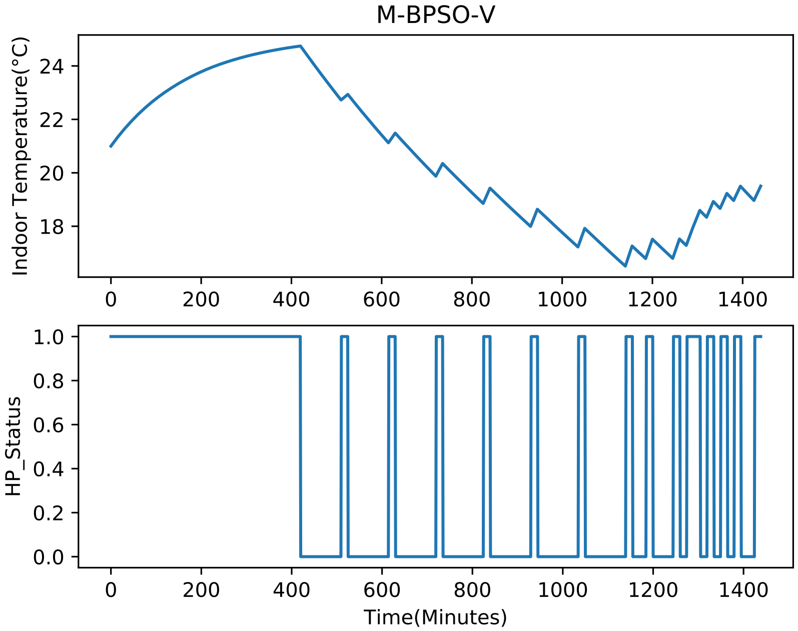


Figure 5.56: Heat pump schedule and corresponding indoor temperature (C/S/15 M-BPSO-V)

As mentioned before, the HP scheduling simulation using M-BPSO-V was executed 50 times and the best results reported. The performance of M-BPSO-V in terms of best obtained cost and minimum indoor temperature over the 50 running times are

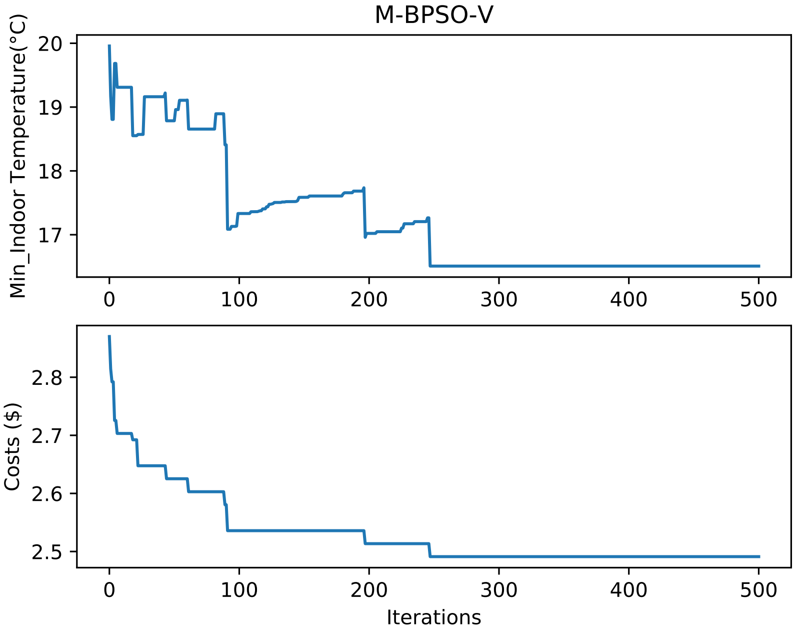


Figure 5.57: Minimum indoor temperature and cost variation (C/S/15 M-BPSO-V)

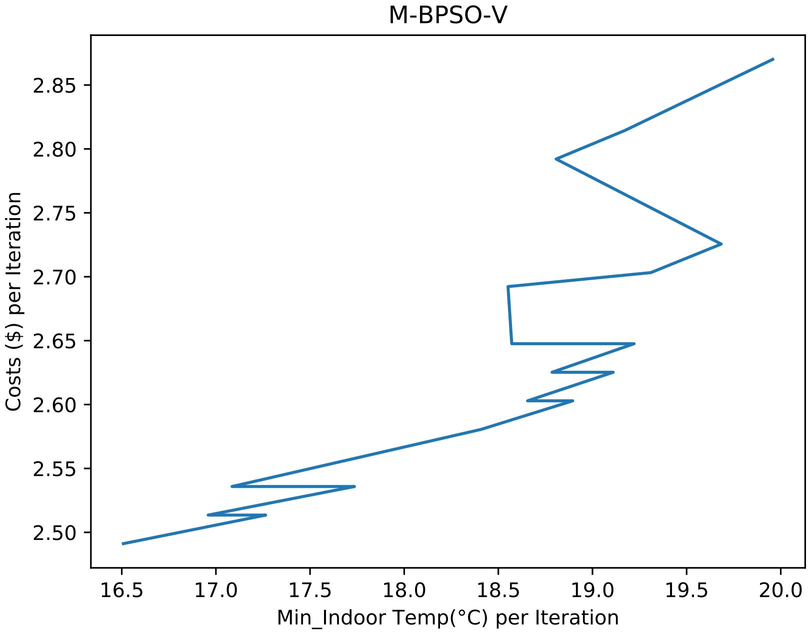
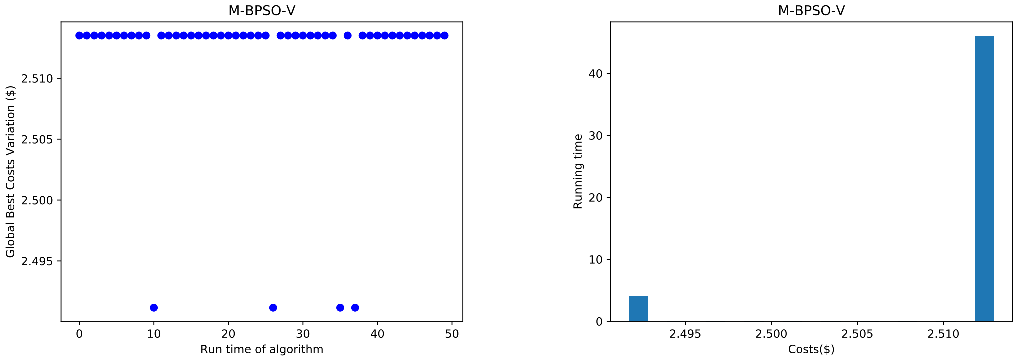


Figure 5.58: Search space trajectory of cost vs. minimum indoor temperature (C/S/15 M-BPSO-V)

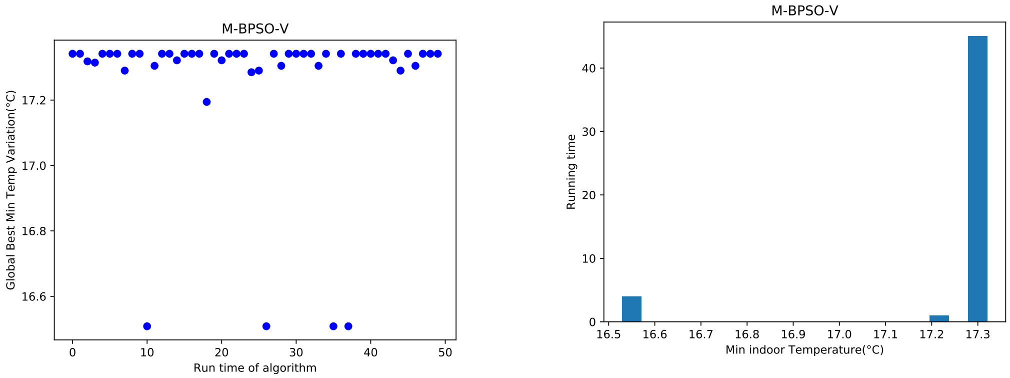
shown in Fig. 5.59 and Fig. 5.60, respectively. The histograms show that in about 10% of runs, the M-BPSO-V converged to global optima and most of the times it converged to a close sub-optimal point. The M-BPSO-V had superior performance in comparison with the other two algorithms, since they were not able to find global optima in all runs.



(a) Costs

(b) Histogram of the costs

Figure 5.59: The best achieved costs during each running time (C/S/15 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.60: The best achieved min temp during each running time (C/S/15 M-BPSO-V)

5.3 Optimization of HP Schedule Considering Occupant Comfort with Dynamic off-time

While in the prior scenarios the maximum allowed continuous off-time of heat pump was defined as a fixed value considering the indoor reference temperature and the minimum acceptable temperature, this scenario use a dynamic approach for determining the continuous off-time of heat pump based on the actual indoor temperature and minimum acceptable temperature. This provides the algorithm more flexibility for scheduling the heat pump running time. In this regard, the allowed continuous off-time penalty is determined from the deviations of the actual indoor temperature and the minimum acceptable indoor temperature, as detailed in chapter 4.

5.3.1 Optimization of HP with 30 minutes time interval resolution (C/D/30)

Since the thermal dynamics of the building in this scenario is same as N/30 scenario, the analyses regarding the heat loss flow and required heat pump running time are not updated and the same values are used. In addition, the time interval resolution and thus the minimum running time of the heat pump is set to 30 minutes.

5.3.1.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.61. Since in this case, the allowed continuous off-time constrain was determined dynamically, the heat pump was not kept off for a long duration of time.

Fig. 5.62 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. Based on the graph, the GA quickly converged to the optimum cost and then significantly improved the minimum indoor temperature over few iterations.

More details of minimum indoor temperature changes versus electricity cost are

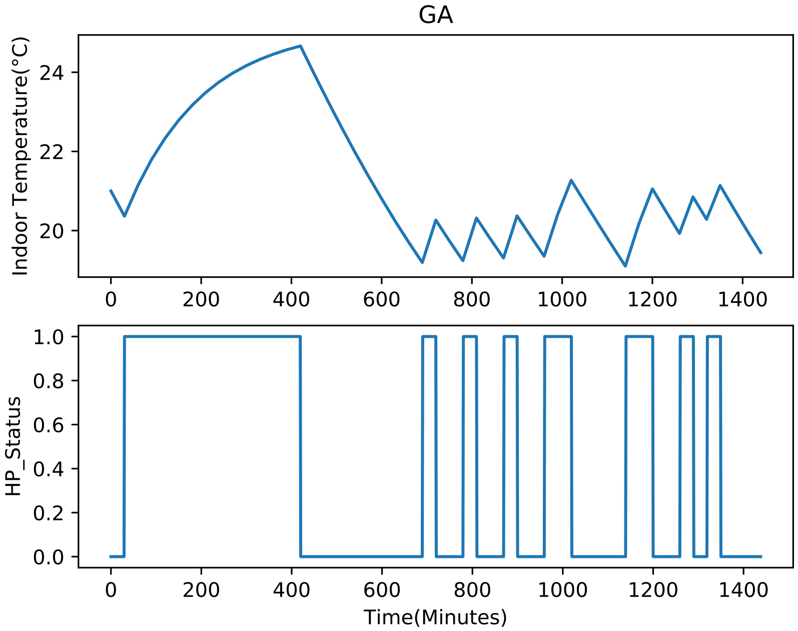


Figure 5.61: Heat pump schedule and corresponding indoor temperature (C/D/30 GA)

provided in Fig. 5.63. It shows that, after converging to the cost optimum, the GA significantly improved minimum indoor temperature during the remaining iterations.

As mentioned before, the HP scheduling simulation using GA was executed 50 times and the best results are reported. The performance of GA in terms of best obtained cost and minimum indoor temperature over the 50 running times are illustrated in Fig. 5.64 and Fig. 5.65, respectively. The histograms show that only in about 10% of runs, the GA converged to global optima and most of the times it converged to a close sub-optimal point.

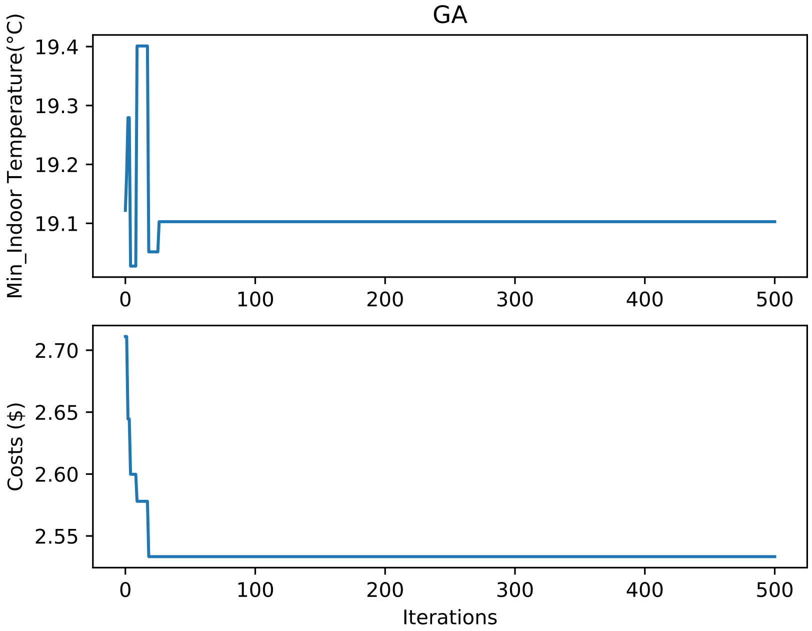


Figure 5.62: Minimum indoor temperature and cost variation (C/D/30 GA)

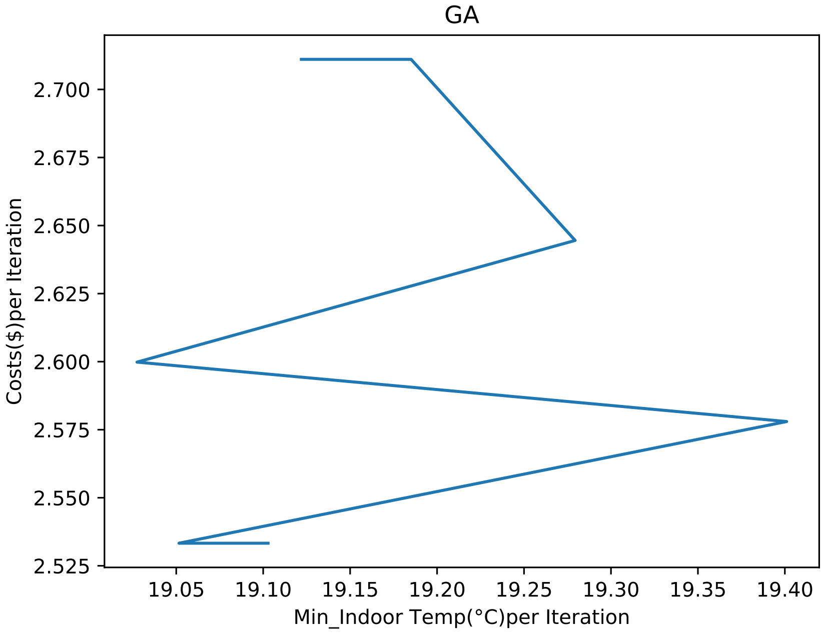
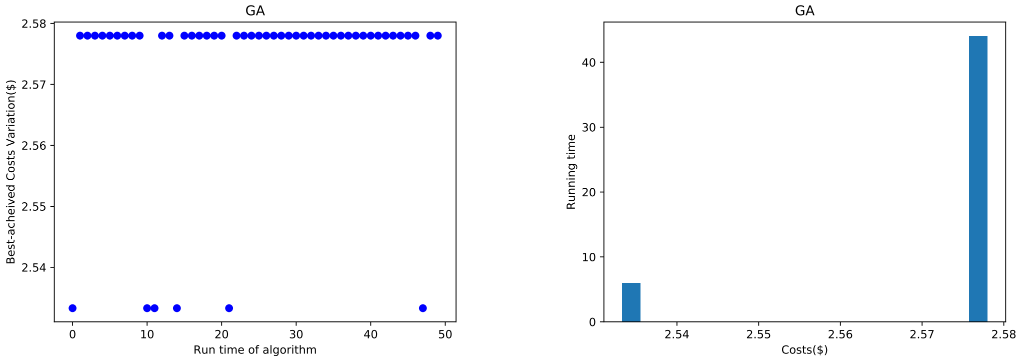


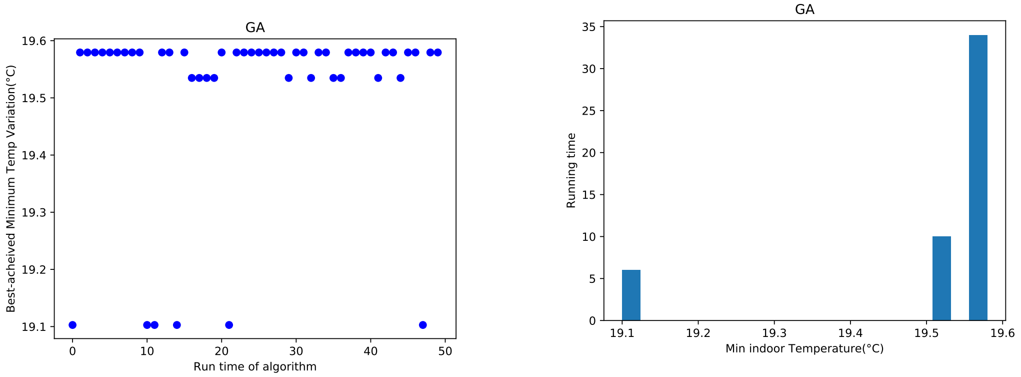
Figure 5.63: Search space trajectory of cost vs. minimum indoor temperature (C/D/30 GA)



(a) Costs

(b) Histogram of the costs

Figure 5.64: The best achieved costs during each run (C/D/30 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.65: The best achieved min temp during each run (C/D/30 GA)

5.3.1.2 Results using M-BPSO-S

Fig. 5.66 shows the heat pump running schedule and indoor temperature corresponding to the best result obtained using M-BPSO-S. The occupant thermal comfort is considered by enforcing the continuous off state of the heat pump, determined dynamically.

Fig. 5.67 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. The algorithm could not achieve to global optima in terms of both the electricity cost and minimum indoor temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.68.

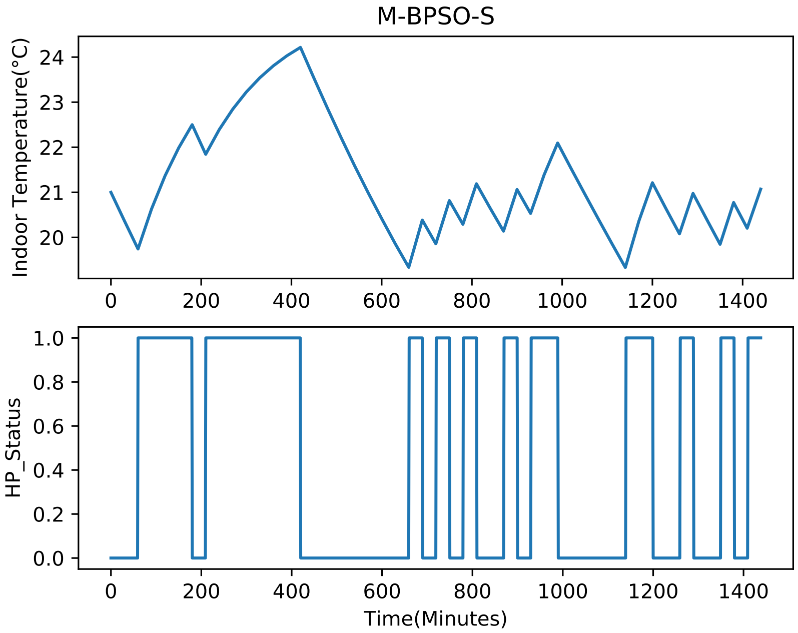


Figure 5.66: Heat pump schedule and corresponding indoor temperature (C/D/30 M-BPSO-S)

Since the scheduling simulation using M-BPSO-S was executed 50 times and the best records are presented, more information regarding the obtained results during whole execution period are presented in Fig. 5.69 and 5.70. The histograms show that the algorithm got trapped in various local optima points during the execution

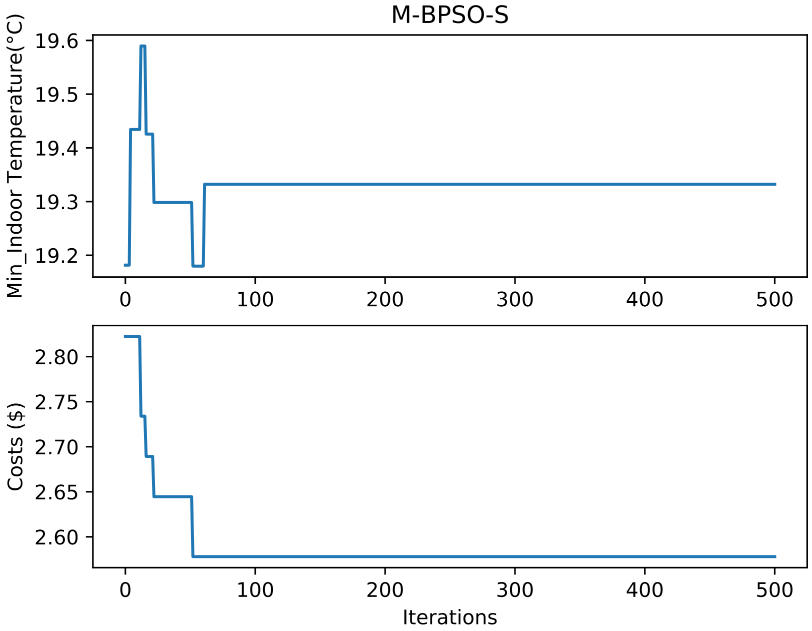


Figure 5.67: Minimum indoor temperature and cost variation (C/D/30 M-BPSO-S)

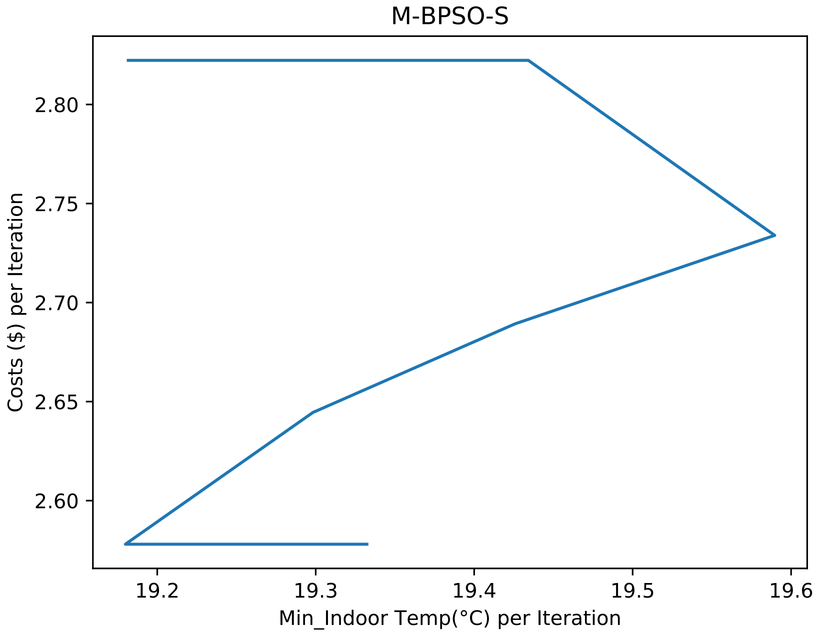
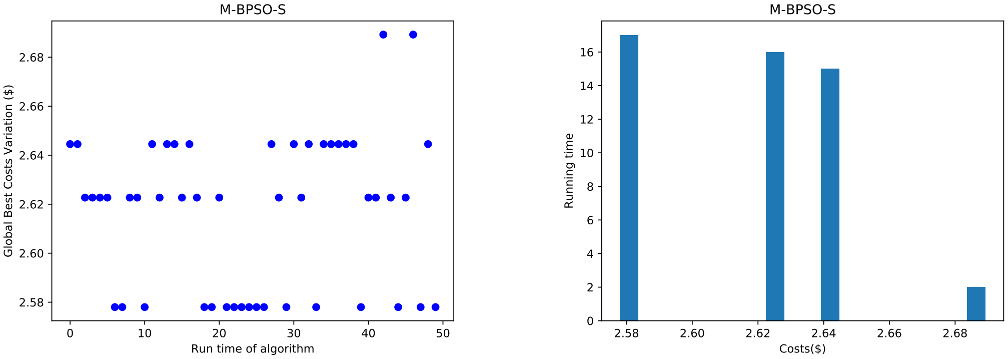


Figure 5.68: Search space trajectory of cost vs. minimum indoor temperature (C/D/30 M-BPSO-S)

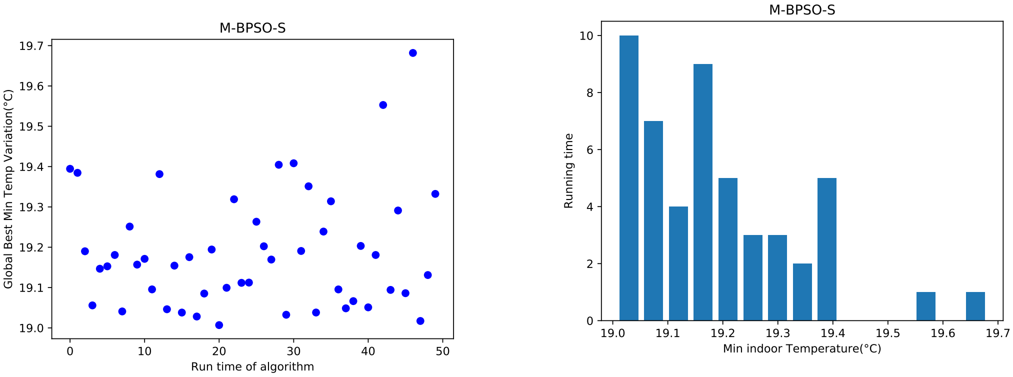
period.



(a) Costs

(b) Histogram of the costs

Figure 5.69: The best achieved costs during each run (C/D/30 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.70: The best achieved min temp during each run (C/D/30 M-BPSO-S)

5.3.1.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the M-BPSO-V are illustrated in Fig. 5.71. Since in this case, the allowed continuous off-time constrain determined dynamically, the heat pump was not kept off for a long duration of time.

Fig. 5.72 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. While the cost optimization leads to temperature decrease, the algorithm successfully increases the minimum indoor temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.73. It shows that, after converging to the cost optimum, the M-BPSO-V significantly improved minimum indoor temperature during the remaining iterations.

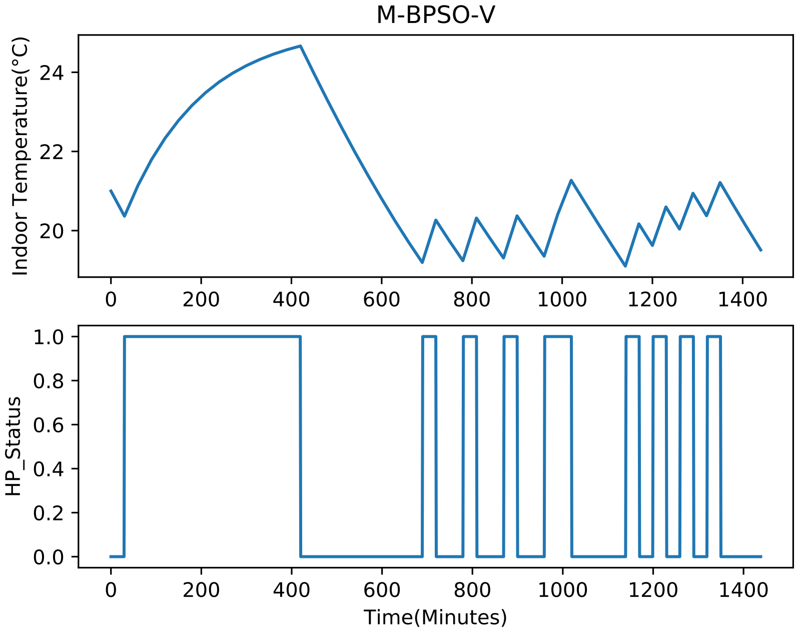


Figure 5.71: Heat pump schedule and corresponding indoor temperature (C/D/30 M-BPSO-V)

As mentioned before, the HP scheduling simulation using M-BPSO-V was executed 50 times and the best results reported. The performance of M-BPSO-V in terms of best obtained cost and minimum indoor temperature over the 50 running times are

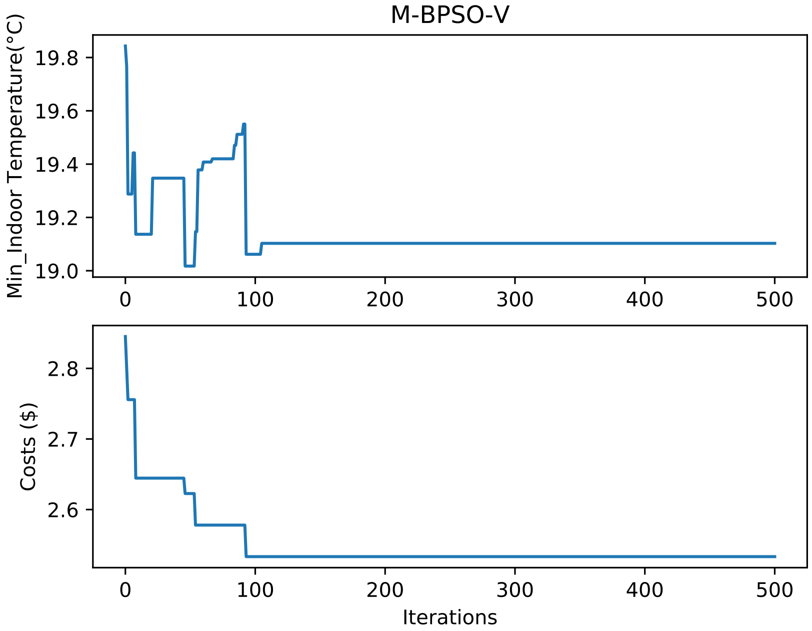


Figure 5.72: Minimum indoor temperature and cost variation (C/D/30 M-BPSO-V)

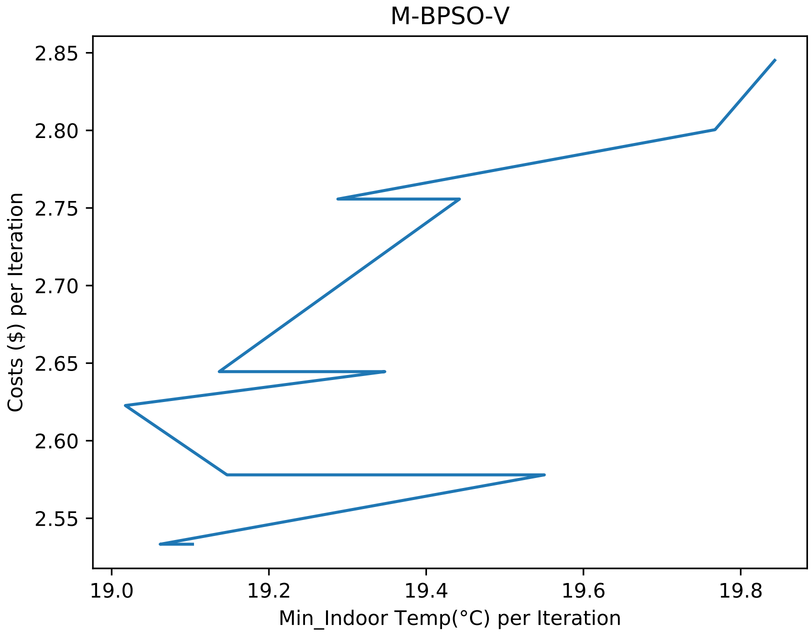
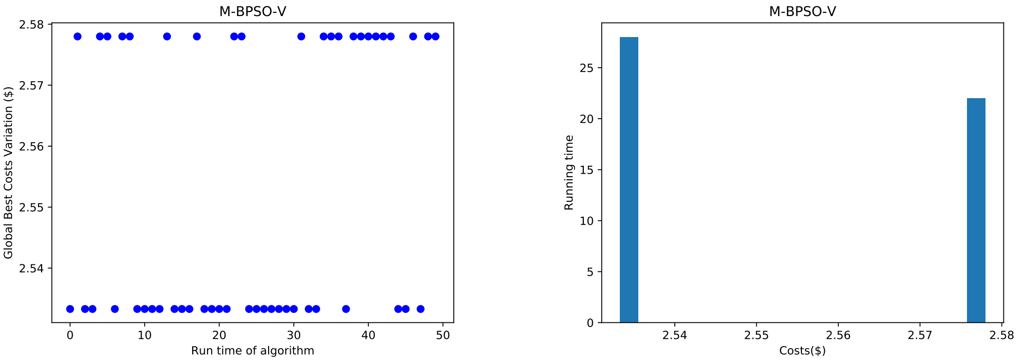


Figure 5.73: Search space trajectory of cost vs. minimum indoor temperature (C/D/30 M-BPSO-V)

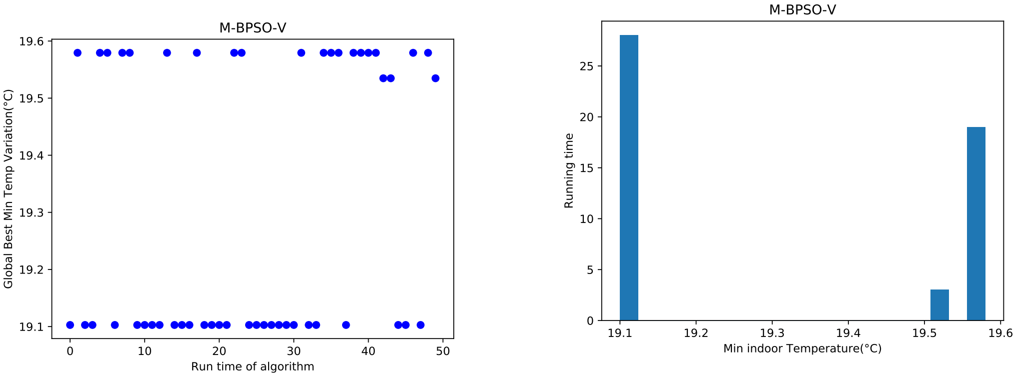
shown in Fig. 5.74 and Fig. 5.74, respectively. The graph shows that M-BPSO-V converged to global optimum more than 25 times which proves its superior performance in comparison with the other algorithms considered in this thesis.



(a) Costs

(b) Histogram of the costs

Figure 5.74: The best achieved costs during each run (C/D/30 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.75: The best achieved min temp during each run (C/D/30 M-BPSO-V)

5.3.2 Optimization of HP with 15 minutes time interval resolution (C/D/15)

In this scenario, the time interval resolution and thus the minimum running time of the heat pump is set to 15 minutes.

Since the thermal dynamics of the building in this scenario is same as N/15 scenario, the analyses regarding the heat loss flow and required heat pump running time are not updated and the same values are used.

5.3.2.1 Results using GA

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the GA are illustrated in Fig. 5.76. Since in this case, the allowed continuous off-time constrain was determined dynamically, the heat pump was not kept off for a long duration of time.

Fig. 5.77 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. Based on the graph, the GA quickly converged to the optimum cost and then significantly improved the minimum indoor temperature over few iterations. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.78. It shows that, after converging to the cost optimum, the GA significantly improved minimum indoor temperature during the remaining iterations.

As mentioned before, the HP scheduling simulation using GA was executed 50 times and the best results are reported. The performance of GA in terms of best obtained cost and minimum indoor temperature over the 50 running times are illustrated in Fig. 5.79 and Fig. 5.80, respectively. The histograms show that the algorithm converged to optimum cost in 12 runs, while in about 5 runs it reached to corresponding optimum temperature.

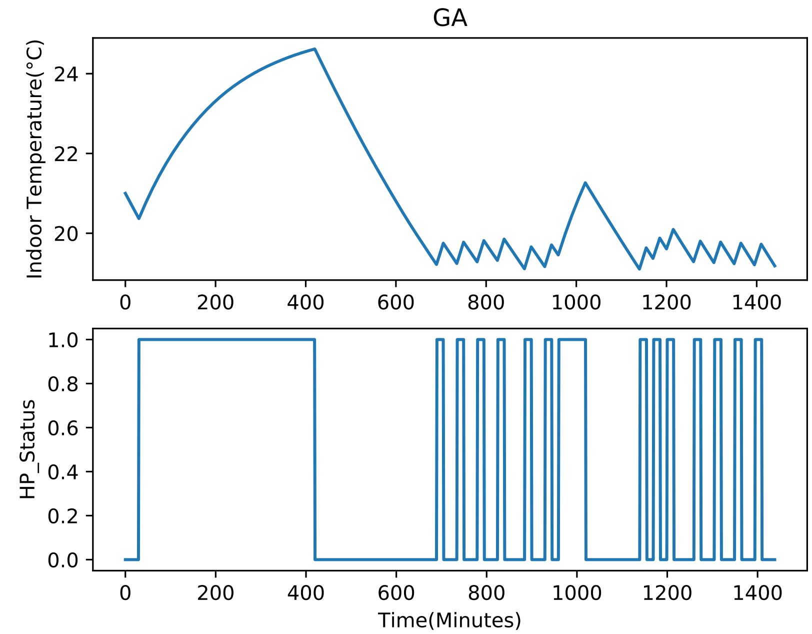


Figure 5.76: Heat pump schedule and corresponding indoor temperature (C/D/15 GA)

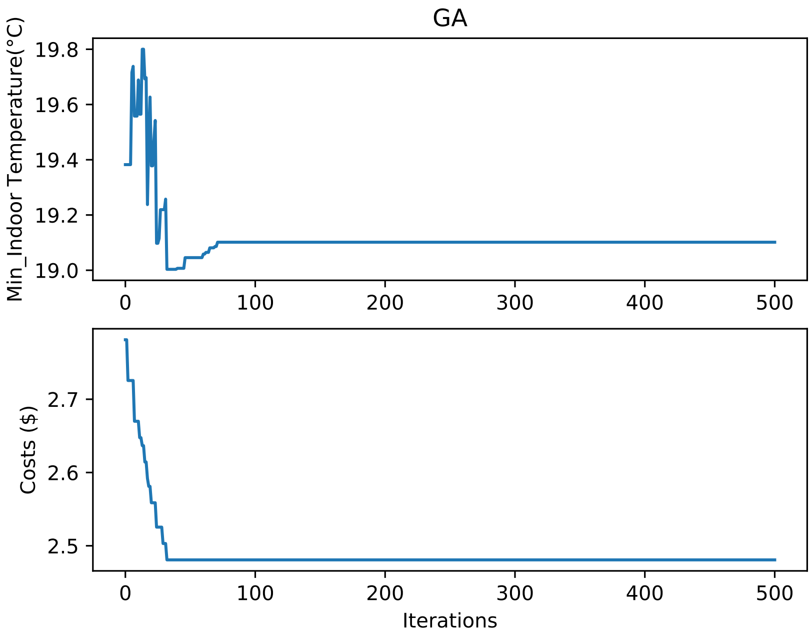


Figure 5.77: Minimum indoor temperature and cost variation (C/D/15 GA)

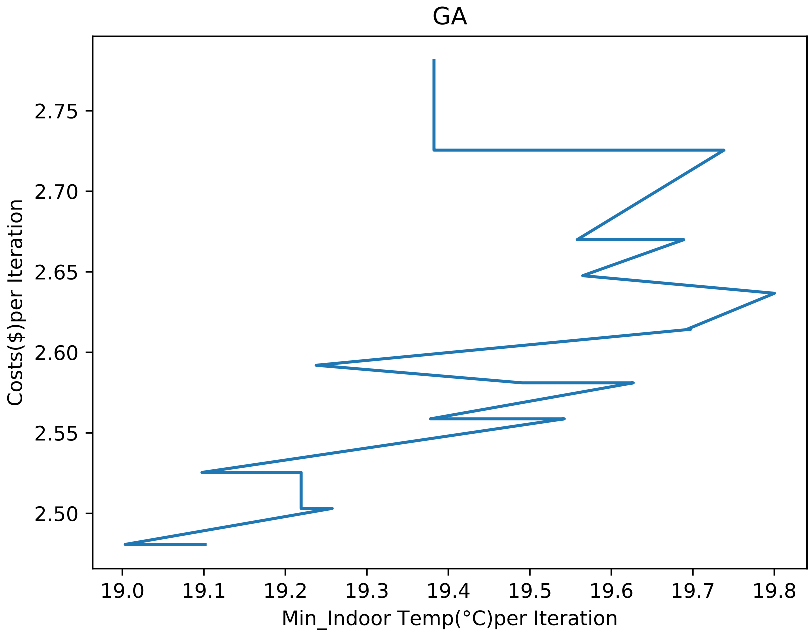
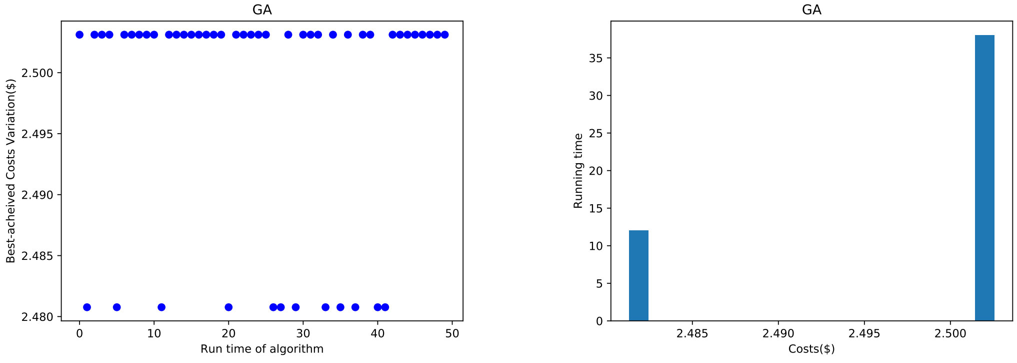


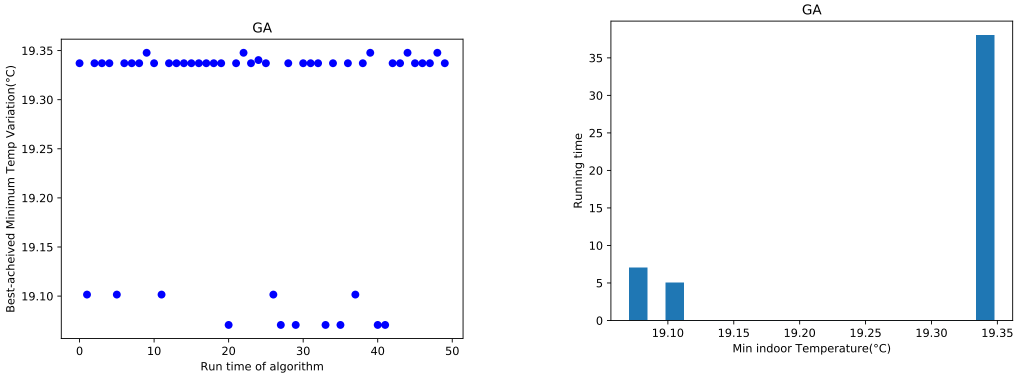
Figure 5.78: Search space trajectory of cost vs. minimum indoor temperature (C/D/15 GA)



(a) Costs

(b) Histogram of the costs

Figure 5.79: C/D/15 GA-The best achieved costs during each run (C/D/15 GA)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.80: C/D/15 GA-The best achieved min temp during each run (C/D/15 GA)

5.3.2.2 Results using M-BPSO-S

Fig. 5.81 shows the heat pump running schedule and indoor temperature corresponding to the best result obtained using M-BPSO-S. The occupant thermal comfort is considered by enforcing the continuous off state of the heat pump, determined dynamically.

Fig. 5.82 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. The graph shows that algorithm did not converge to optimum cost and corresponding minimum indoor temperature. More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.83.

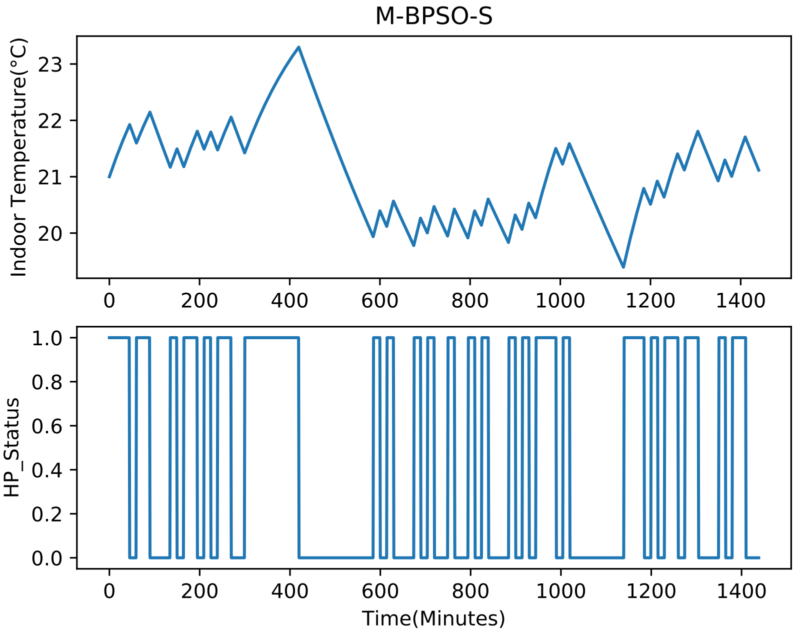


Figure 5.81: Heat pump schedule and corresponding indoor temperature (C/D/15 M-BPSO-S)

Since the scheduling simulation using M-BPSO-S was executed for 50 times and the best records are presented, more information regarding the obtained results during whole execution period are presented in Fig. 5.84 and Fig. 5.85. The histograms show that the algorithm got trapped in various local optima points during the execution period.

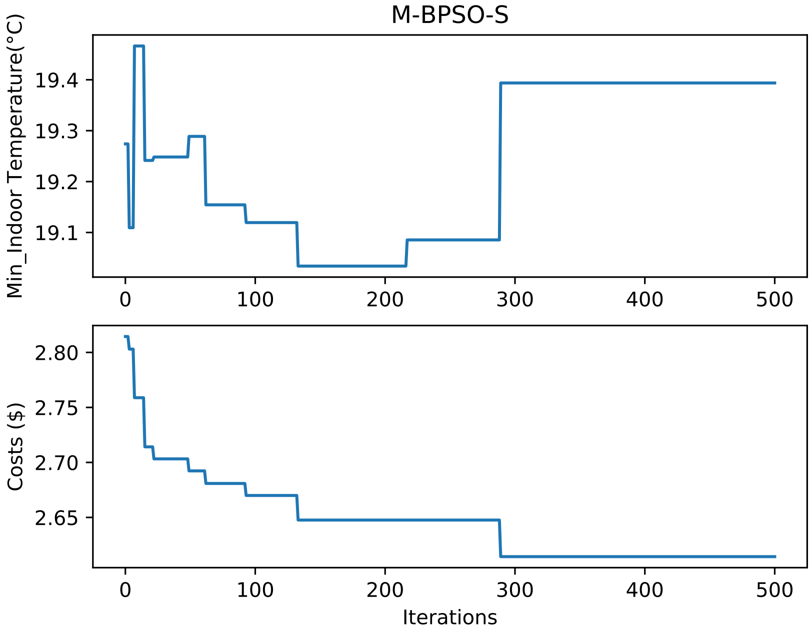


Figure 5.82: Minimum indoor temperature and cost variation (C/D/15 M-BPSO-S)

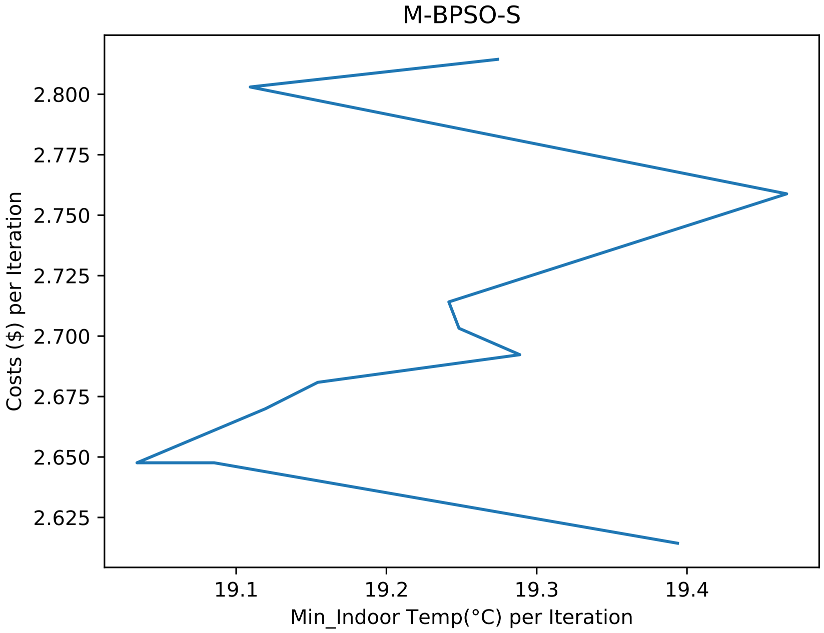
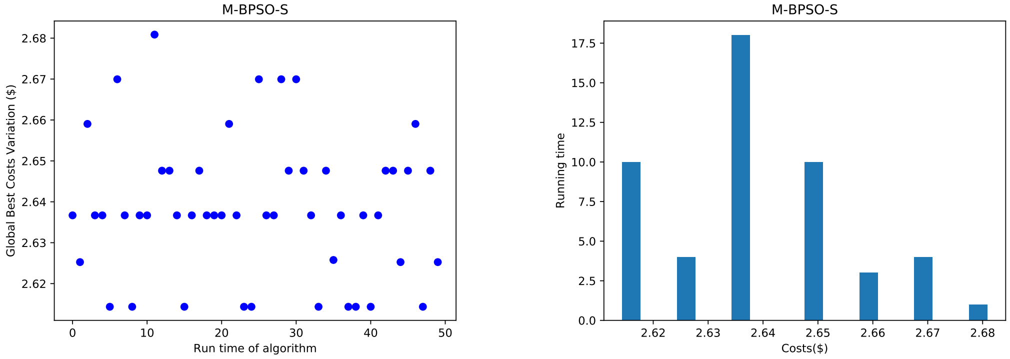


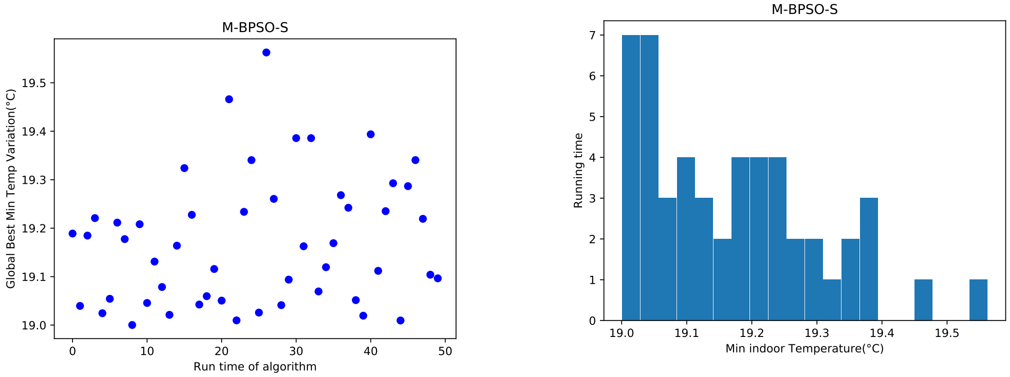
Figure 5.83: Search space trajectory of cost vs. minimum indoor temperature (C/D/15 M-BPSO-S)



(a) Costs

(b) Histogram of the costs

Figure 5.84: The best achieved costs during each run (C/D/15 M-BPSO-S)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.85: The best achieved min temp during each run (C/D/15 M-BPSO-S)

5.3.2.3 Results using M-BPSO-V

The heat pump running schedule and indoor temperature profile corresponding to the best result obtained using the M-BPSO-V are illustrated in Fig. 5.86. Since in this scenario, the allowed continuous off-time constrain was determined dynamically, the heat pump was not kept off for a long duration of time.

Fig. 5.87 illustrates the variation of minimum indoor temperature and costs over the course of 500 iterations. Based on the graph, the M-BPSO-V converged to the optimum cost in about 200 iterations and then significantly improved the minimum indoor temperature over few iterations.

More details of minimum indoor temperature changes versus electricity cost are provided in Fig. 5.88. It shows that, after converging to the cost optimum, the M-BPSO-V significantly improved minimum indoor temperature during the remaining simulation time.

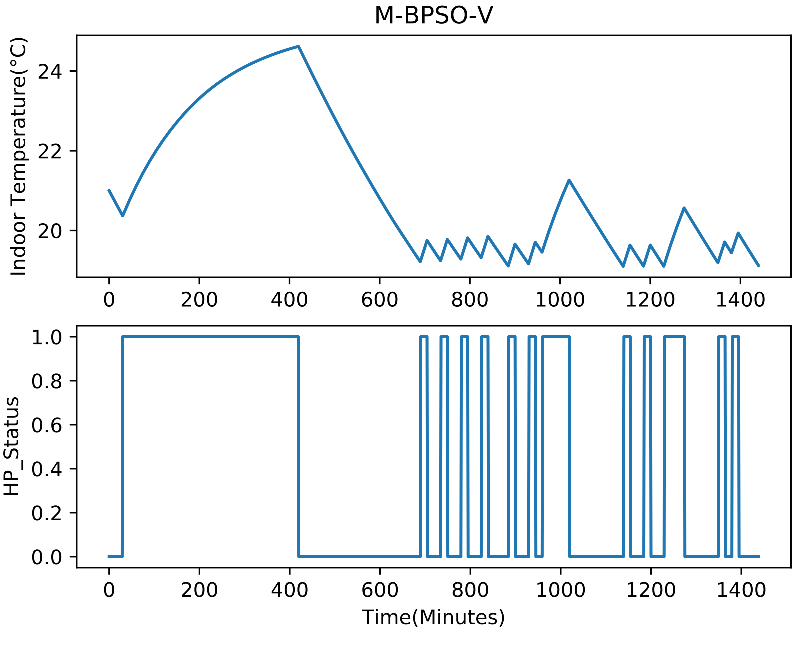


Figure 5.86: Heat pump schedule and corresponding indoor temperature (C/D/15 M-BPSO-V)

As mentioned before, the HP scheduling simulation using M-BPSO-V was executed

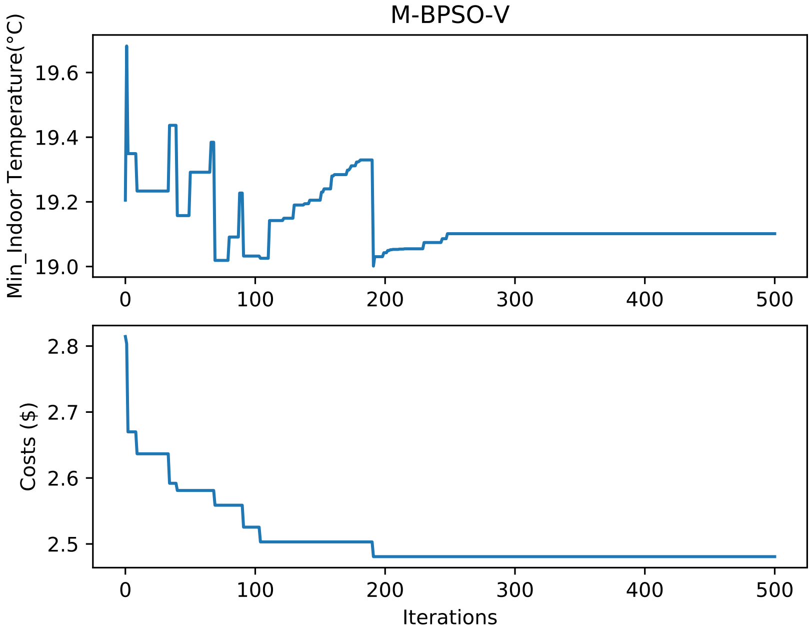


Figure 5.87: Minimum indoor temperature and cost variation (C/D/15 M-BPSO-V)

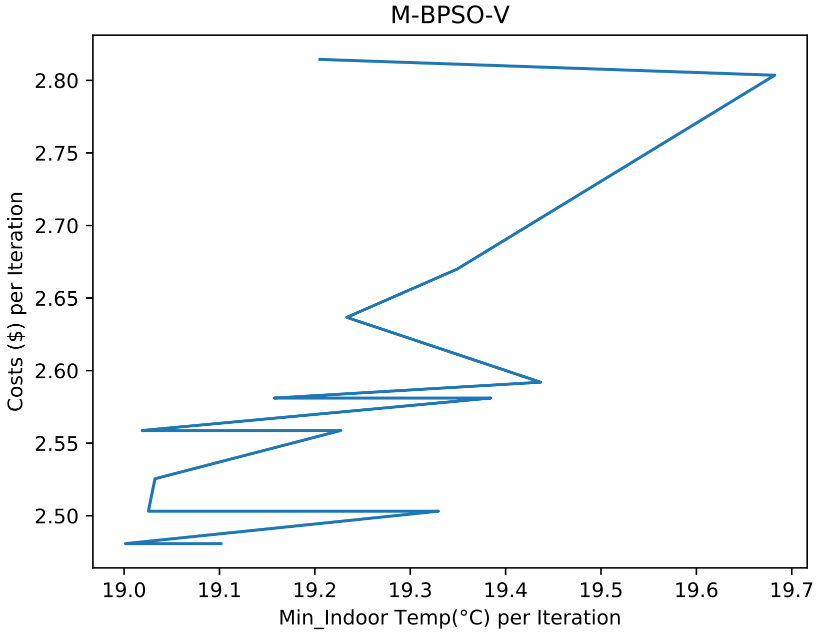
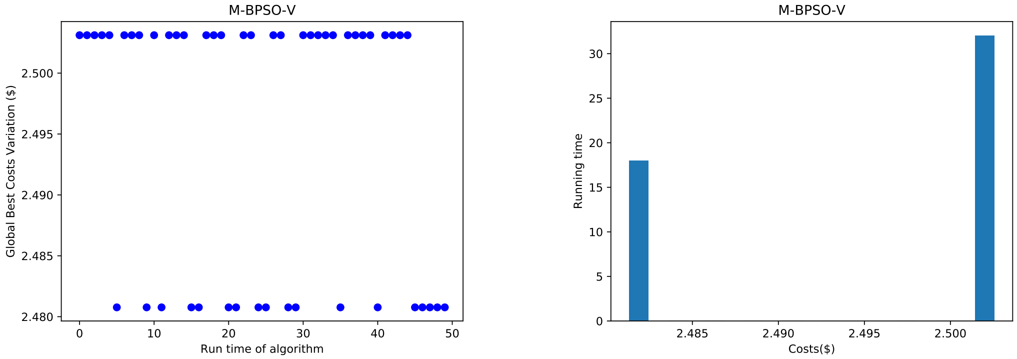


Figure 5.88: Search space trajectory of cost vs. minimum indoor temperature (C/D/15 M-BPSO-V)

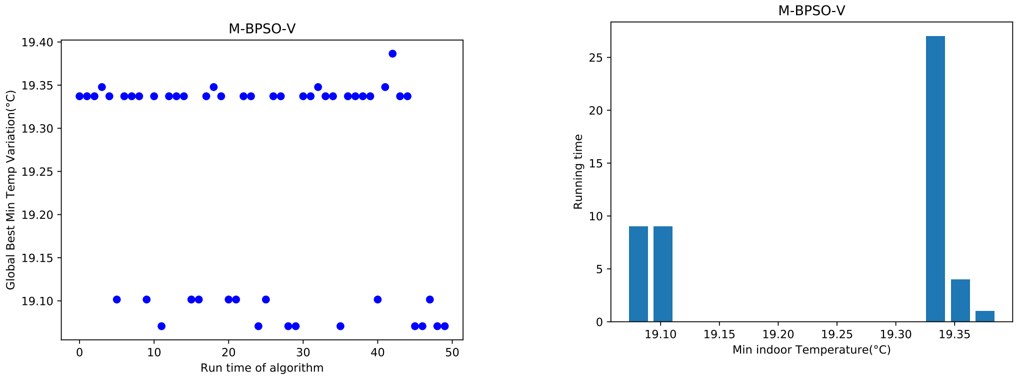
50 times and the best results are reported. The performance of M-BPSO-V in terms of best obtained cost and minimum indoor temperature over the 50 running times are illustrated in Fig. 5.89 and Fig. 5.90, respectively. Graph shows that in more than 15 runs the algorithm converged to optimum cost, while in only 9 runs reached to corresponding optimum temperature. As seen, in most runs the algorithm converged to a close sub-optimal point.



(a) Costs

(b) Histogram of the costs

Figure 5.89: The best achieved costs during each running time (C/D/15 M-BPSO-V)



(a) Minimum temperature

(b) Histogram of the minimum temperature

Figure 5.90: The best achieved min temp during each running time (C/D/15 M-BPSO-V)

5.4 Analysis

Analysis regarding the results obtained by the three algorithms in the six scenarios are provided in this section. In these simulation experiments, the performance of algorithms are evaluated. In addition, the influence of time resolution and the static vs. dynamic approach regarding the determination of maximum allowed continuous off-time of heat pump on the scheduling problem are investigated.

5.4.1 Algorithm Performance

5.4.1.1 N/30 Scenario

The results obtained by the algorithms in the N/30 scenario in which the thermal comfort of occupants was not considered are summarized in Table 5.1. The table indicates that minimization of electricity cost can lead to significant drop of indoor temperature, negatively effecting the thermal comfort of the occupants. The results show that GA and M-BPSO-V reached to global optima (cost of 2.3098$ and minimum indoor temperature of 12.93°C), but M-BPSO-S did not converge to the global optima point. Considering the NFEs and iteration number for first visit of global optima, the GA reaches the optima in smaller number of steps than M-BPSO-V. In particular, the GA converged to the global optima in 23 iterations or after 14300 NFEs (11.50s), while M-BPSO-V took 31 iterations or 19100 NFEs (7.40s). Since all algorithms were executed using the same computing resources, the wall time can be considered as a fair comparison parameter. The table proves that M-BPSO-V is faster than GA since its wall time records (total time and first visit of the optimum point) are significantly shorter than GA.

Table 5.1: Summary of simulation results in the N/30 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.3098	2.3544	2.3098
Minimum Temperature (°C)	12.93	14.48	12.93
Temperature Avg. (°C)	19.83	20.00	19.90
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	335.37	117.60	124.61
NFE for first visit of the global optima	14300	-	19100
Iteration No. for first visit of the global optima	23	-	31
Wall time for first visit of global the optima (s)	11.50	-	7.40

5.4.1.2 N/15 Scenario

The results obtained by each algorithm in the N/15 scenario (thermal comfort of occupants was not considered) are presented in Table 5.2. Similarly to the the N/30 scenario, the electricity cost reduction leads to significant temperature decrease with a negative effect on the thermal comfort of occupants. The table shows that GA and M-BPSO-V converged to global optima (cost of 2.2572$ and minimum indoor temperature of 12.98 °C), but M-BPSO-S could not reach this point during the whole optimization progress. The GA converged to optimum point after 80 iterations or 48500 NFEs (62.21s) while M-BPSO-V achieved this point in 103 iterations or 62300 NFEs (40.61s). Considering the NFEs and number of iteration for first visit of global optima, GA reaches the optima in smaller number of steps than M-BPSO-V. However, the total wall time and the wall time for first visiting of global optima indicate that M-BPSO-V is the fastest algorithm. Since the algorithms are executed using the same computing resources, the wall time would be a fair comparison parameter.

Table 5.2: Summary of simulation results in the N/15 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.2572	2.4469	2.2572
Minimum Temperature (°C)	12.98	16.44	12.98
Temperature Avg. (°C)	19.78	20.71	19.76
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	492.73	189.82	208.28
NFE for first visit of the global optima	48500	-	62300
Iteration No. for first visit of the global optima	80	-	103
Wall time for first visit of the global optima (s)	62.21	-	40.61

5.4.1.3 C/S/30 Scenario

To address the negative effect of cost optimization on thermal comfort of occupants, both electricity cost and indoor temperature variations must be considered in the heat pump scheduling problem. Hence, in this scenario, the second objective is enforced through the fixed maximum continuous off-state of the heat pump. The summary of simulation results obtained by GA, M-BPSO-S and M-BPSO-V in the C/S/30 scenario are provided in Table 5.3. The results indicate that GA and M-BPSO-V achieved to global optima point (2.7775$, 19.44°C). However, the M-BPSO-S algorithm converged to optimum cost, but it could not achieve the optima on the second objective during the optimization period. The table shows that GA converged to global optimum in 21 iterations or 13100 NFEs (11.47s). In comparison, the M-BPSO-V algorithm reached to this point within 93 iterations or 56300 NFEs (22.05s). At first glance, the performance of GA is faster than the M-BPSO-V algorithm. However, GA does not provide consistent results, as shown in Fig. 5.34 and Fig. 5.35. The graphs show that during 50 runs, GA got several times trapped in local optima points, while M-BPSO-V successfully converged to global optima in every execution shown in Fig. 5.44 and Fig. 5.45. In addition, the total wall time shows that M-BPSO-V algorithm is generally faster than GA.

Table 5.3: Summary of simulation results in the C/S/30 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.7775	2.7775	2.7775
Minimum Temperature (°C)	19.44	19.16	19.44
Temperature Avg. (°C)	21.32	21.08	21.36
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	333.82	113.97	122.01
NFE for first visiting of global optima	13100	-	56300
Iteration No. for first visiting of global optima	21	-	93
Wall time for first visiting of global optima (s)	11.47	-	22.05

5.4.1.4 C/D/30 Scenario

The simulation results achieved by all algorithms in the C/D/30 scenario are summarized in Table 5.4. The results show that GA and M-BPSO-V algorithm reached the global optimal point (2.5332$, 19.10°C). However, M-BPSO-S only converged to values close to the optima. The table shows that the performance of GA in terms of fast convergence to global optima is better than M-BPSO-V algorithm. Since the table shows the best results achieved by each algorithm after 50 executions, more detailed analysis is required to compare the performance of the optimization algorithms. As mentioned earlier, the details regarding the performance of GA in each execution are illustrated in Fig. 5.64 and Fig. 5.65. They show that GA converged to global optimum cost and temperature in 6 times. In comparison, the analysis of the M-BPSO-V results (Fig. 5.74 and Fig. 5.75) indicate that M-BPSO-V converged to optimum cost and temperature more than 25 times. Therefore, it can be concluded that M-BPSO-V algorithm is more consistent than GA.

Table 5.4: Summary of simulation results in the C/D/30 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.5332	2.5779	2.5332
Minimum Temperature (°C)	19.10	19.33	19.10
Temperature Avg. (°C)	21.27	21.23	21.26
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	322.84	99.63	100.58
NFE for first visit of the global optima	16100	-	63500
Iteration No. for first visit of the global optima	26	-	105
Wall time for first visit of the global optima (s)	11.93	-	20.51

5.4.1.5 C/S/15 Scenario

The simulation results obtained by all algorithms in the C/S/15 scenario are illustrated in Table 5.5. The results show that only M-BPSO-V algorithm reached the global optima point (2.4911$, 16.50°C). Although, GA and M-BPSO-S did not converge to the optima, their results are close to optimum value. The performance of GA is better than M-BPSO-S since its results are relatively closer to the global optima.

Table 5.5: Summary of simulation results in the C/S/15 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.5135	2.6138	2.4911
Minimum Temperature (°C)	17.34	18.12	16.50
Temperature Avg. (°C)	20.80	20.72	20.60
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	534.14	235.81	240.99
NFE for first visit of the global optima	-	-	148700
Iteration No. for first visit of the global optima	-	-	247
Wall time for first visit of the global optima (s)	-	-	116.89

5.4.1.6 C/D/15 Scenario

To mitigate the negative effect of cost minimization on occupants comfort in terms of indoor temperature, the second objective (indoor temperature) is enforced to optimization problem through maximum allowed continuous off-time constraint of heat pump based on dynamic approach. The simulation results achieved by all algorithms in the C/D/15 scenario are illustrated in Table 5.6. The results show that GA and M-BPSO-V algorithm reached the global optima point (2.4807$, 19.10°C). Although, M-BPSO-S did not converge to optima, its best result is close to optimum value. The table shows that the performance of GA in terms of fast convergence to global optima is better than for M-BPSO-V algorithm. The table shows the best results achieved by each algorithm after 50 execution times. As mentioned earlier, details about the performance of GA algorithm in each execution are shown in Fig. 5.79 and Fig. 5.80. They show that while GA converged to optimum cost 12 times, it only achieved to optimum corresponding minimum indoor temperature 5 times. Thus, the global optima convergence rate of GA is 10% for this problem. In comparison, the analysis of the M-BPSO-V results shown in Fig. 5.89 and Fig. 5.90 indicate that M-BPSO-V converged to optimum cost in 18 times while in 9 cases it also reached corresponding optimal minimum indoor temperature. As a result, the global optima convergence rate of M-BPSO-V is 18%, or about twice that of GA. The results also confirm that in the cases where the algorithms do not reach the global optima point, their results are still close to the optimum values.

Table 5.6: Summary of simulation results in the C/D/15 scenario

Results	**GA**	**MBPSO-S**	**MBPSO-V**
Cost($)	2.4807	2.6143	2.4807
Minimum Temperature (°C)	19.10	19.40	19.10
Temperature Avg. (°C)	21.05	21.08	21.06
Total NFEs (-)	300500	300500	300500
Total iterations (-)	500	500	500
Total Wall time (s)	487.23	194.78	203.00
NFE for first visit of the global optima	43100	-	149300
Iteration No. for first visit of the global optima	71	-	248
Wall time for first visit of the global optima (s)	55.41	-	99.16

According to the presented analysis and evaluations, the performance of M-BPSO-V was superior to the other two algorithms in terms of consistency, ability to converge to global optima in all scenarios, fast convergence and short execution time.

5.4.2 Time resolution and off-time methods

To analyse the influence of time resolution and different off-time determination approaches (static and dynamic) in heat pump scheduling optimization, the results of the best performed algorithm (M-BPSO-V) in different scenarios are compared. Table 5.7 shows the results of M-BPSO-V algorithm in all six scenarios. According to the table, increasing the time resolution led to more cost reduction which is due to the more accurate mapping between required running time of the heat pump and its required running time-slots. In particular, the required running time-slots of heat pump in the scenarios with 30 minutes time interval resolution is equal to 22, while this number in scenarios with 15 minutes time interval resolution is 43. Hence, the reduction of the duration of heat pump operation can bring more savings.

While higher time resolution leads to higher cost savings, it could have negative effect on the thermal comfort delivered to occupants. According to the table and results regarding the C/S/30 and C/S/15 scenarios in which the thermal comfort is enforced through static off-time of heat pump, the use of higher time resolution leads to about 3°C drop of minimum indoor temperature. Hence, the algorithm could not satisfy the thermal comfort in C/S/15 scenario. While the maximum allowed continuous off-time in C/S/30 and C/S/15 scenarios is equal, the total running time duration of heat pump in C/S/15 scenario is shorter than C/S/30 due to more efficient mapping between operating time and time-slots. Therefore, the thermal discomfort in C/S/15 is caused by two reasons: shorter running time of heat pump and the static off-time approach which do not provide sufficient flexibility for the algorithm to benefit from the thermal storage capacity of the building.

The results in C/S/30 and C/D/30 indicate that the use of dynamic off-time approach can lead to more cost reduction while maintaining the thermal comfort of occupants. The dynamic approach provides more flexibility for the algorithm to take advantage of thermal storage capacity of the building. As a result, the algorithm can efficiently use the time intervals with higher electricity price while maintaining the thermal comfort delivered to occupants. The results achieved in C/D/15 scenario indicate that the dynamic approach successfully addressed the thermal discomfort caused by higher time resolution while benefiting from its cost reduction capability.

It should be mentioned that higher time resolution leads to longer wall time, iterations or NFEs for the first visit of the global optima. This is due to the increase of the number of dimensions which makes the optimization problem more complex. In particular, while the number of dimensions in the scenarios with 15 minutes time interval resolution is two times greater than the scenarios with 30 minutes resolution, their complexity is greater in power of two.

Table 5.7: Summary of M-BPSO-V results in the 6 considered scenarios

Results	**N/30**	**N/15**	**C/S/30**	**C/D/30**	**C/S/15**	**C/D/15**
Cost($)	2.3098	2.2572	2.7775	2.5332	2.4911	2.4807
Minimum Temperature (°C)	12.93	12.98	19.44	19.10	16.50	19.10
Temperature Avg. (°C)	19.90	19.76	21.36	21.26	20.60	21.06
Total NFEs (-)	300500	300500	300500	300500	300500	300500
Total iterations (-)	500	500	500	500	500	500
Total Wall time (s)	124.61	208.28	122.01	100.58	240.99	203
NFE for first visit of the global optima	19100	62300	56300	63500	148700	149300
Iteration No. for first visit of the global optima	31	103	93	105	247	248
Wall time for first visit of the global optima (s)	7.40	40.61	22.05	20.51	116.89	99.16

Chapter 6

Conclusions and Future Work

This thesis presents an intelligent approach for scheduling the heat pump in a residential building using three metaheuristic optimization algorithms: the GA, M-BPSO-S and M-BPSO-V. The M-BPSO-S and M-BPSO-V algorithms represent an original contribution of this thesis to improve the performance of the original BPSO algorithm. The performance of all algorithms is evaluated through a number of simulation experiments. Accordingly, the scheduling simulations are executed for two main scenarios, and several sub-scenarios. While the main scenarios determine consideration or negligence of the thermal comfort delivered to occupants, sub-scenarios distinguish the simulation time resolution and methodology (static or dynamic) used for characterizing the allowed continuous off-time of the heat pump. These scenarios are classified as follows

- Optimization of HP schedule not considering occupant comfort
    - Optimization of HP with 30 minutes time interval resolution (N/30)
    - Optimization of HP with 15 minutes time interval resolution (N/15)
- Optimization of HP schedule considering occupant comfort with static off-time
    - Optimization of HP with 30 minutes time interval resolution (C/S/30)
    - Optimization of HP with 15 minutes time interval resolution (C/S/15)

- Optimization of HP schedule considering occupant comfort with dynamic off-time
    - Optimization of HP with 30 minutes time interval resolution (C/D/30)
    - Optimization of HP with 15 minutes time interval resolution (C/D/15)

The obtained simulation results confirmed that the proposed approach can successfully optimize the heat pump operation scheduling. According to the provided analysis, the proposed M-BPSO-V algorithm had the best performance in terms of the consistency, global optimality and the speed of convergence. The M-BPSO-S results indicate that adding the mutation operator to original BPSO could not, on its own, address all shortcomings of the standard BPSO. Therefore, another modification of the search procedure was implemented resulting in M-BPSO-V algorithm. In this algorithm, the sigmoid transfer function is superseded with a V-shaped function. The transfer function is the main part of BPSO that maps the continuous search space to discrete binary space. The results show that new transfer function and its position updating rules significantly improve the performance of the BPSO algorithm.

The best GA run reached the global optimum in a smaller number of NFEs and iterations than M-BPSO-V. However, it did not perform consistently well across all runs. In addition, the wall time to reach the optimum for M-BPSO-V was shorter than that of GA in most runs, and the total wall time of M-BPSO-V algorithm runs was significantly shorter in all scenarios. For some scenarios, such as C/S/15, the GA did not find the global optima at all, while M-BPSO-V did. This was despite the fact that the diversity of GA population was increased through the use of a combination of a single point and double point crossover operators. This confirms that, among the optimization approaches examined in this thesis, the M-BPSO-V algorithm introduced in this thesis offers the best performance for solving the heat pump scheduling problem. For ease of comparison, additional plots regarding the performance of the algorithms examined in this thesis are provided in the appendix.

To demonstrate the quality of the optimization results, consider the following quantitative analysis of the process. The optimal results obtained by GA and M-BPSO-V in N/30 and C/D/15 scenarios show that, by spending extra 17 cents, the minimum indoor temperature can be increased by 6.17°C and reach the value of 19.10 °C. Further analysis of the results obtained by the algorithm proposed in this thesis, M-BPSO-V, confirms that increasing the time resolution can lead higher cost reduction, but at the cost of reduced thermal comfort. The analysis of the static vs. dynamic off-time determination indicates that dynamic approach can bring more cost reduction while maintaining the thermal comfort of occupants. Employing a higher time resolution with the dynamic off-time approach leads to the highest cost reduction while maintaining the thermal comfort.

Future research on the intelligent heat pump scheduling problem may take a number of different avenues. First, the approaches examined in this thesis can be further enhanced to improve their overall performance in solving this specific problem. Another possibility is to examine the working schedule of other appliances and to take into account the internal heat gain. More advanced options include consideration of the occupancy status of the house or prediction of the activities of the occupants.

Bibliography

[1] J. Barton, S. Huang, D. Infield, M. Leach, D. Ogunkunle, J. Torriti, and M. Thomson, "The evolution of electricity demand and the role for demand side participation, in buildings and transport," *Energy Policy*, vol. 52, pp. 85–102, 2013.

[2] N. Grid, *Uk future energy scenarios*, 2014.

[3] D. Six, J. Desmedt, D Vahnoudt, and J. Bael, "Exploring the flexibility potential of residential heat pumps combined with thermal energy storage for smart grids," in *21th International Conference on Electricity Distribution, Paper*, vol. 442, 2011.

[4] J. L. Cremer, M. Pau, F. Ponci, and A. Monti, "Optimal scheduling of heat pumps for power peak shaving and customers thermal comfort.," in *SMARTGREENS*, 2017, pp. 23–34.

[5] P. Palensky and D. Dietrich, "Demand side management: Demand response, intelligent energy systems, and smart loads," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 3, pp. 381–388, 2011, ISSN: 1941-0050. DOI: 10.1109/TII.2011.2158841.

[6] M. Pau, J. L. Cremer, F. Ponci, and A. Monti, "Impact of customers flexibility in heat pumps scheduling for demand side management," in *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*, 2017, pp. 1–6. DOI: 10.1109/EEEIC.2017.7977681.

[7] M. H. Albadi and E. F. El-Saadany, "Demand response in electricity markets: An overview," in *2007 IEEE Power Engineering Society General Meeting*, 2007, pp. 1–5. DOI: 10.1109/PES.2007.385728.

[8] N. Chapman, L. Zhang, N. Good, and P. Mancarella, "Exploring flexibility of aggregated residential electric heat pumps," in *2016 IEEE International Energy Conference (ENERGYCON)*, 2016, pp. 1–6. DOI: 10.1109/ENERGYCON.2016.7514082.

[9] K. Bettgenhäuser, M. Offermann, T. Boermans, M. Bosquet, J. Grözinger, B. von Manteuffel, and N. Surmeli, "Heat pump implementation scenarios until 2030," *Ecofys, Koln, Germany*, 2013.

[10] C. Molitor, F. Ponci, A. Monti, D. Cali, and D. Müller, "Consumer benefits of electricity-price-driven heat pump operation in future smart grids," in *2011 IEEE International Conference on Smart Measurements of Future Grids (SMFG) Proceedings*, IEEE, 2011, pp. 75–78.

[11] J. Love, A. Z. Smith, S. Watson, E. Oikonomou, A. Summerfield, C. Gleeson, P. Biddulph, L. F. Chiu, J. Wingfield, C. Martin, A. Stone, and R. Lowe, "The addition of heat pump electricity load profiles to gb electricity demand: Evidence from a heat pump field trial," *Applied Energy*, vol. 204, pp. 332 –342, 2017, ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2017.07.026.

[12] M. Akmal and B. Fox, "Modelling and simulation of underfloor heating system supplied from heat pump," in *2016 UKSim-AMSS 18th International Conference on Computer Modelling and Simulation (UKSim)*, 2016, pp. 246–251. DOI: 10.1109/UKSim.2016.13.

[13] N. Good, L. Zhang, A. Navarro-Espinosa, and P. Mancarella, "Physical modeling of electro-thermal domestic heating systems with quantification of economic and environmental costs," in *Eurocon 2013*, 2013, pp. 1164–1171. DOI: 10.1109/EUROCON.2013.6625128.

[14] L. Zhang, N. Good, A. Navarro-Espinosa, and P. Mancarella, "Modelling of household electro-thermal technologies for demand response applications," in *IEEE PES Innovative Smart Grid Technologies, Europe*, 2014, pp. 1–6. DOI: 10.1109/ISGTEurope.2014.7028917.

[15] A. Arteconi, N. Hewitt, and F. Polonara, "Domestic demand-side management (dsm): Role of heat pumps and thermal energy storage (tes) systems," *Applied Thermal Engineering*, vol. 51, no. 1, pp. 155 –165, 2013, ISSN: 1359-4311. DOI: https://doi.org/10.1016/j.applthermaleng.2012.09.023. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1359431112006357.

[16] S. Klein, W. Beckman, J. Mitchel, J. Duffie, N. Duffie, T. Freeman, *et al.*, "Trnsys manual," *University of Wisconsin*, 1998.

[17] M. Diekerhof, S. Vorkampf, and A. Monti, "Distributed optimization algorithm for heat pump scheduling," in *IEEE PES Innovative Smart Grid Technologies, Europe*, 2014, pp. 1–6. DOI: 10.1109/ISGTEurope.2014.7028943.

[18] M. Diekerhof, S. Schwarz, and A. Monti, "Distributed optimization for electro-thermal heating units," in *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2016, pp. 1–6. DOI: 10.1109/ISGTEurope.2016.7856258.

[19] C. Molitor, F. Ponci, A. Monti, D. Cali, and D. Müller, "Consumer benefits of electricity-price-driven heat pump operation in future smart grids," in *2011 IEEE International Conference on Smart Measurements of Future Grids (SMFG) Proceedings*, 2011, pp. 75–78. DOI: 10.1109/SMFG.2011.6125774.

[20] M. Loesch, D. Hufnagel, S. Steuer, T. Faßnacht, and H. Schmeck, "Demand side management in smart buildings by intelligent scheduling of heat pumps," in *2014 IEEE International Conference on Intelligent Energy and Power Systems (IEPS)*, 2014, pp. 1–6. DOI: 10.1109/IEPS.2014.6874181.

[21] B. P. Bhattarai, B. Bak-Jensen, J. R. Pillai, and M. Maier, "Demand flexibility from residential heat pump," in *2014 IEEE PES General Meeting — Conference Exposition*, 2014, pp. 1–5. DOI: 10.1109/PESGM.2014.6939348.

[22] L. Zhang, N. Chapman, N. Good, and P. Mancarella, "Exploiting electric heat pump flexibility for renewable generation matching," in *2017 IEEE Manchester PowerTech*, 2017, pp. 1–6. DOI: 10.1109/PTC.2017.7981266.

[23] F. De Angelis, M. Boaro, D. Fuselli, S. Squartini, F. Piazza, and Q. Wei, "Optimal home energy management under dynamic electrical and thermal constraints," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1518–1527, 2013, ISSN: 1941-0050. DOI: 10.1109/TII.2012.2230637.

[24] M. Pau, J. L. Cremer, F. Ponci, and A. Monti, "Day-ahead scheduling of electric heat pumps for peak shaving in distribution grids," in *Smart Cities, Green Technologies, and Intelligent Transport Systems*, B. Donnellan, C. Klein, M. Helfert, O. Gusikhin, and A. Pascoal, Eds., Cham: Springer International Publishing, 2019, pp. 27–51, ISBN: 978-3-030-02907-4.

[25] D. Banks, *An introduction to thermogeology: ground source heating and cooling.* John Wiley & Sons, 2012.

[26] M. Zogg, "History of heat pumps-swiss contributions and international milestones," in *9th International IEA Heat Pump Conference, Zürich, Switzerland*, 2008, pp. 20–22.

[27] C. Wu, *Thermodynamics and heat powered cycles: a cognitive engineering approach.* Nova Publishers, 2007.

[28] O. Kramer, *Genetic algorithm essentials.* Springer, 2017, vol. 679.

[29] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based systems*, vol. 89, pp. 228–249, 2015.

[30] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135 –4151, 2011, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2011.02.032. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494611000962.

[31] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82 –117, 2013, Prediction, Control and Diagnosis using Advanced Neural Computations, ISSN: 0020-0255. DOI: 10.1016/j.ins.2013.02.041. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025513001588.

[32] A. Gogna and A. Tayal, “Metaheuristics: Review and application,” *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 25, no. 4, pp. 503–526, 2013. DOI: 10.1080/0952813X.2013.782347. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-84887827270&doi=10.1080%2f0952813X.2013.782347&partnerID=40&md5=093719d8c0b7a2fdb49aea418ac6393f.

[33] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[34] L. Bianchi, M. Dorigo, L. M. Gambardella, and W. J. Gutjahr, “A survey on metaheuristics for stochastic combinatorial optimization,” *Natural Computing*, vol. 8, no. 2, pp. 239–287, 2009.

[35] H. H. Hoos and T. Stützle, *Stochastic local search: Foundations and applications*. Elsevier, 2004.

[36] S. Kirkpatrick, “Optimization by simulated annealing: Quantitative studies,” *Journal of statistical physics*, vol. 34, no. 5-6, pp. 975–986, 1984.

[37] G. Cornuéjols, “Valid inequalities for mixed integer linear programs,” *Mathematical Programming*, vol. 112, no. 1, pp. 3–44, 2008.

[38] M. Avriel, *Nonlinear programming: analysis and methods*. Courier Corporation, 2003.

[39] A. H. Land and A. G. Doig, “An automatic method for solving discrete programming problems,” in *50 Years of Integer Programming 1958-2008*, Springer, 2010, pp. 105–132.

[40] A. R. Simpson, G. C. Dandy, and L. J. Murphy, “Genetic algorithms compared to other techniques for pipe optimization,” *Journal of water resources planning and management*, vol. 120, no. 4, pp. 423–443, 1994.

[41] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[42] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Advances in engineering software*, vol. 69, pp. 46–61, 2014.

[43] J. H. Holland, “Genetic algorithms,” *Scientific american*, vol. 267, no. 1, pp. 66–73, 1992.

[44] I. Rechenberg, “Evolutionsstrategien,” in *Simulationsmethoden in der Medizin und Biologie*, B. Schneider and U. Ranft, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1978, pp. 83–114, ISBN: 978-3-642-81283-5.

[45] J. R. Koza and J. R. Koza, *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992, vol. 1.

[46] D. Simon, “Biogeography-based optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008, ISSN: 1941-0026. DOI: 10.1109/TEVC.2008.919004.

[47] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[48] V. Černỳ, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *Journal of optimization theory and applications*, vol. 45, no. 1, pp. 41–51, 1985.

[49] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Gsa: A gravitational search algorithm," *Information sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[50] R. Formato, *Central force optimization: A new metaheuristic with applications in applied electromagnetics. prog electromagn res 77: 425–491*, 2007.

[51] O. K. Erol and I. Eksin, "A new optimization method: Big bang–big crunch," *Advances in Engineering Software*, vol. 37, no. 2, pp. 106–111, 2006.

[52] H. Shah-Hosseini, "Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation," *International Journal of Computational Science and Engineering*, vol. 6, no. 1-2, pp. 132–140, 2011.

[53] G. Beni and J. Wang, "Swarm intelligence in cellular robotic systems," in *Robots and biological systems: towards a new bionics?* Springer, 1993, pp. 703–712.

[54] E. Bonabeau, M. Dorigo, D. d. R. D. F. Marco, G. Theraulaz, G. Théraulaz, *et al.*, *Swarm intelligence: from natural to artificial systems*, 1. Oxford university press, 1999.

[55] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2006.

[56] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, IEEE, vol. 4, 1995, pp. 1942–1948.

[57] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (abc) algorithm," *Journal of global optimization*, vol. 39, no. 3, pp. 459–471, 2007.

[58] L. J. Fogel, A. J. Owens, and M. J. Walsh, "Artificial intelligence through simulated evolution," 1966.

[59] F. Glover, "Tabu search—part i," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989. DOI: 10.1287/ijoc.1.3.190. [Online]. Available: https://doi.org/10.1287/ijoc.1.3.190.

[60] F. Glover, "Tabu search—part ii," *ORSA Journal on Computing*, vol. 2, no. 1, pp. 4–32, 1990. DOI: 10.1287/ijoc.2.1.4. [Online]. Available: https://doi.org/10.1287/ijoc.2.1.4.

[61] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," 2007, pp. 4661–4667. DOI: 10.1109/CEC.2007.4425083. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-78149290542&doi=10.1109%2fCEC.2007.4425083&partnerID=40&md5=08d4de3994a464b64f2de7884ac78761.

[62] F. Ramezani and S. Lotfi, “Social-based algorithm (sba),” *Applied Soft Computing*, vol. 13, no. 5, pp. 2837 –2856, 2013, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2012.05.018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494612002542.

[63] A. Kaveh, “Colliding bodies optimization,” in *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Cham: Springer International Publishing, 2014, pp. 195–232, ISBN: 978-3-319-05549-7. DOI: 10.1007/978-3-319-05549-7_7. [Online]. Available: https://doi.org/10.1007/978-3-319-05549-7_7.

[64] A. Kaveh and V. Mahdavi, “Colliding bodies optimization: A novel metaheuristic method,” *Computers Structures*, vol. 139, pp. 18 –27, 2014, ISSN: 0045-7949. DOI: https://doi.org/10.1016/j.compstruc.2014.04.005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045794914000935.

[65] Y. Tan and Y. Zhu, “Fireworks algorithm for optimization,” in *Advances in Swarm Intelligence*, Y. Tan, Y. Shi, and K. C. Tan, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 355–364.

[66] M. Eita and M. Fahmy, “Group counseling optimization,” *Applied Soft Computing*, vol. 22, pp. 585 –604, 2014, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2014.03.043. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494614001628.

[67] M. A. Eita and M. M. Fahmy, “Group counseling optimization: A novel approach,” in *Research and Development in Intelligent Systems XXVI*, M. Bramer, R. Ellis, and M. Petridis, Eds., London: Springer London, 2010, pp. 195–208, ISBN: 978-1-84882-983-1.

[68] O. Olorunda and A. P. Engelbrecht, “Measuring exploration/exploitation in particle swarms using swarm diversity,” in *2008 IEEE congress on evolutionary computation (IEEE world congress on computational intelligence)*, IEEE, 2008, pp. 1128–1134.

[69] E. Alba and B. Dorronsoro, “The exploration/exploitation tradeoff in dynamic cellular genetic algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 2, pp. 126–142, 2005, ISSN: 1941-0026. DOI: 10.1109/TEVC.2005.843751.

[70] L. Lin and M. Gen, “Auto-tuning strategy for evolutionary algorithms: Balancing between exploration and exploitation,” *Soft Computing*, vol. 13, no. 2, pp. 157–168, 2009.

[71] S. Mirjalili and S. Z. M. Hashim, “A new hybrid psogsa algorithm for function optimization,” in *2010 international conference on computer and information application*, IEEE, 2010, pp. 374–377.

[72] S. Mirjalili, S. Z. M. Hashim, and H. M. Sardroudi, "Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm," *Applied Mathematics and Computation*, vol. 218, no. 22, pp. 11125 –11 137, 2012, ISSN: 0096-3003. DOI: https://doi.org/10.1016/j.amc.2012.04.069. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300312004857.

[73] C. Darwin's, "On the origin of species," *John Murray, London*, 1859.

[74] G. Syswerda, "Simulated crossover in genetic algorithms," in *Foundations of genetic algorithms*, vol. 2, Elsevier, 1993, pp. 239–255.

[75] X.-S. Yang, *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010.

[76] T. Weise, "Global optimization algorithms-theory and application," *Self-Published Thomas Weise*, 2009.

[77] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43. DOI: 10.1109/MHS.1995.494215.

[78] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, 1942–1948 vol.4. DOI: 10.1109/ICNN.1995.488968.

[79] J. Kennedy, "The particle swarm: Social adaptation of knowledge," in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, 1997, pp. 303–308. DOI: 10.1109/ICEC.1997.592326.

[80] J. Dai and J. Ying, "Adaptive mutation based particle swarm optimization algorithm," in *Proceedings of 2012 UKACC International Conference on Control*, 2012, pp. 404–408. DOI: 10.1109/CONTROL.2012.6334664.

[81] S. Sengupta, S. Basak, and R. A. Peters, "Particle swarm optimization: A survey of historical and recent developments with hybridization perspectives," *Machine Learning and Knowledge Extraction*, vol. 1, no. 1, pp. 157–191, 2019.

[82] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 1998, pp. 69–73. DOI: 10.1109/ICEC.1998.699146.

[83] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, V. W. Porto, N. Saravanan, D. Waagen, and A. E. Eiben, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 591–600, ISBN: 978-3-540-68515-9.

[84] S. Cheng, H. Lu, X. Lei, and Y. Shi, "A quarter century of particle swarm optimization," *Complex & Intelligent Systems*, vol. 4, no. 3, pp. 227–239, 2018.

[85] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No.00TH8512)*, vol. 1, 2000, 84–88 vol.1. DOI: 10.1109/CEC.2000.870279.

[86] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, 1999, 1951–1957 Vol. 3. DOI: 10.1109/CEC.1999.785513.

[87] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 1, pp. 58–73, 2002, ISSN: 1941-0026. DOI: 10.1109/4235.985692.

[88] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, vol. 5, 1997, 4104–4108 vol.5. DOI: 10.1109/ICSMC.1997.637339.

[89] S. Mirjalili and A. Lewis, "S-shaped versus v-shaped transfer functions for binary particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.

[90] Zhuangkuo Li and Ning Li, "A novel multi-mutation binary particle swarm optimization for 0/1 knapsack problem," in *2009 Chinese Control and Decision Conference*, 2009, pp. 3042–3047. DOI: 10.1109/CCDC.2009.5192838.

[91] G.-C. Luh, C.-Y. Lin, and Y.-S. Lin, "A binary particle swarm optimization for continuum structural topology optimization," *Applied Soft Computing*, vol. 11, no. 2, pp. 2833 –2844, 2011, The Impact of Soft Computing for the Progress of Artificial Intelligence, ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2010.11.013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494610002905.

[92] *Ontario energy board sets new electricity prices for households and small businesses.* [Online]. Available: https://www.oeb.ca/newsroom/2019/ontario-energy-board-sets-new-electricity-prices-households-and-small-businesses.

[93] M. Z. Degefa, M. Lehtonen, K. Nixon, and M. McCulloch, "A high resolution model of residential internal heat gain — the subtle interdependencies among residential end uses," in *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*, 2015, pp. 1–6. DOI: 10.1109/ISGT-Asia.2015.7386990.

[94] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "Bgsa: Binary gravitational search algorithm," *Natural Computing*, vol. 9, no. 3, pp. 727–745, 2010.

Appendix A: Complementary Plots for Simulation Results

A.1 N/30 Scenario

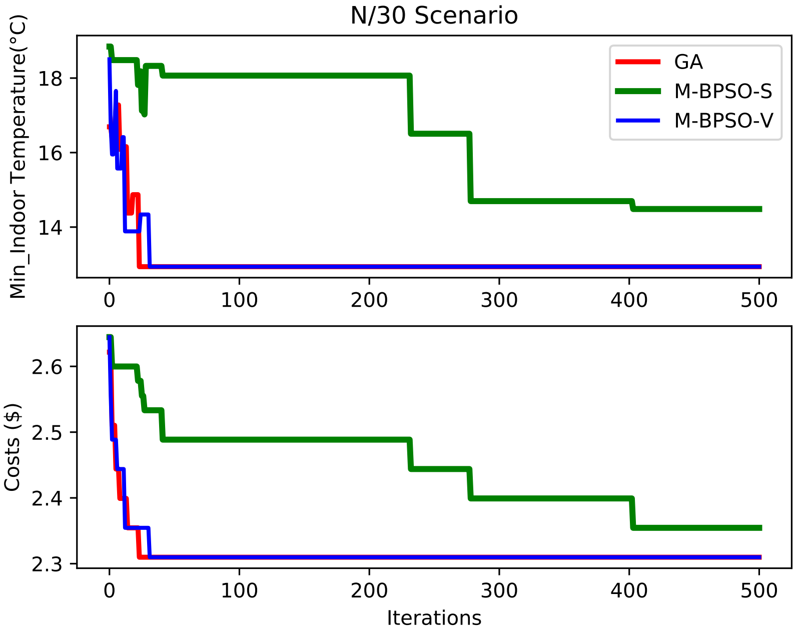


Figure A.1: Minimum indoor temperature and cost variation (N/30)

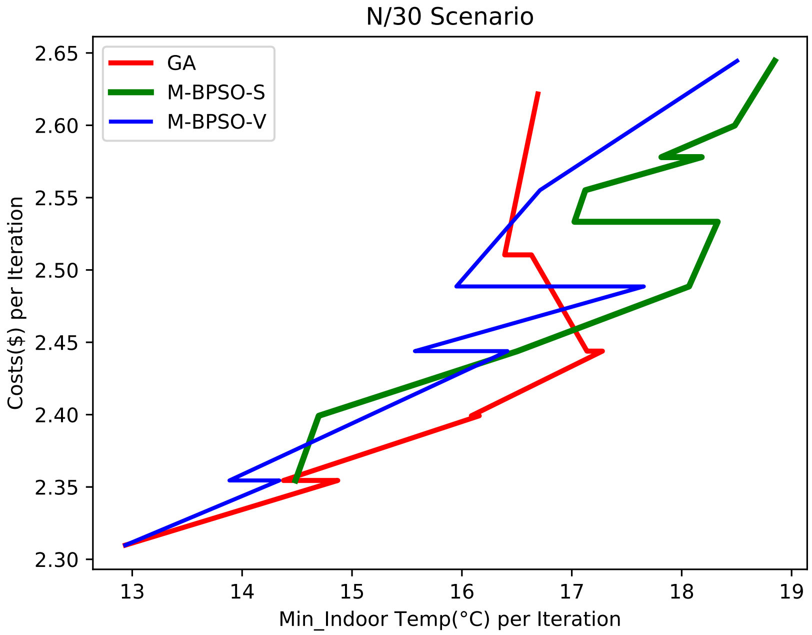


Figure A.2: Search space trajectory (N/30)

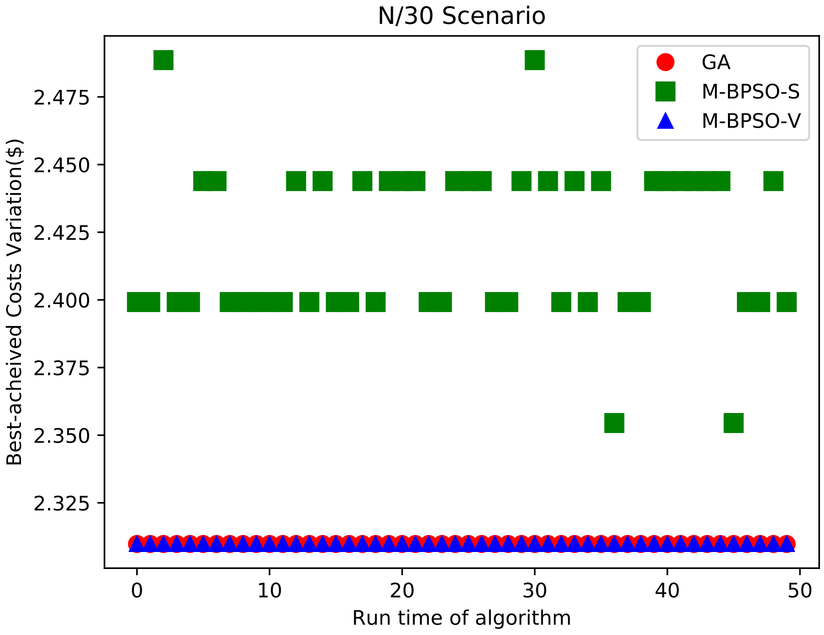


Figure A.3: The best achieved costs during each run (N/30)

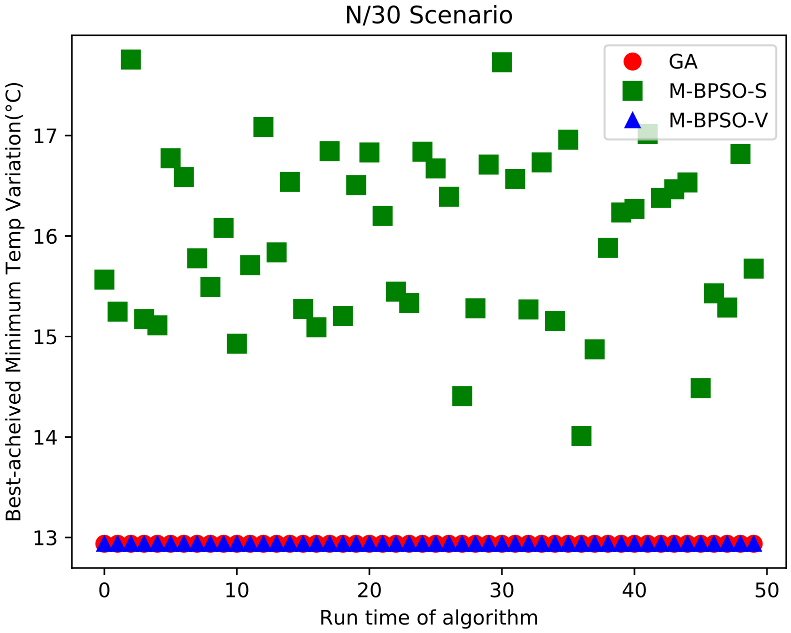


Figure A.4: The best achieved min temp during each run (N/30)

A.2 N/15 Scenario

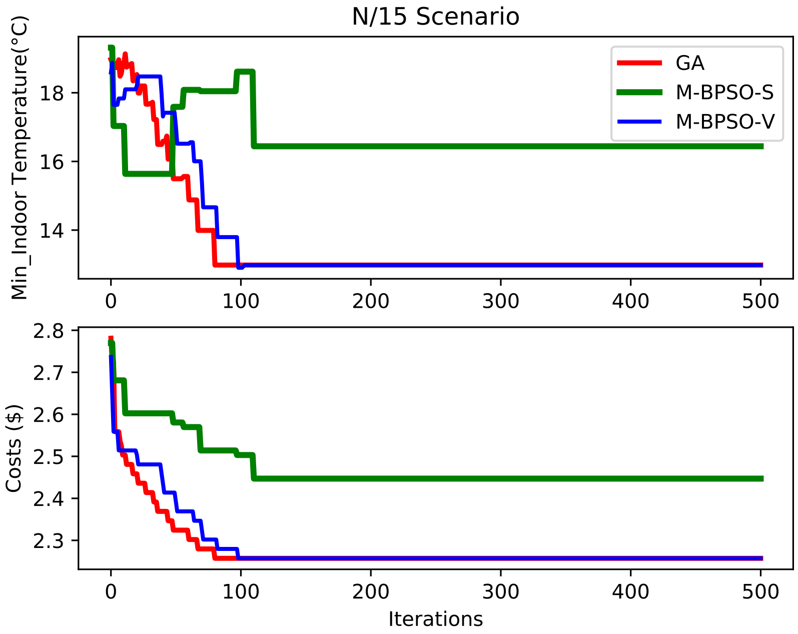


Figure A.5: Minimum indoor temperature and cost variation (N/15)

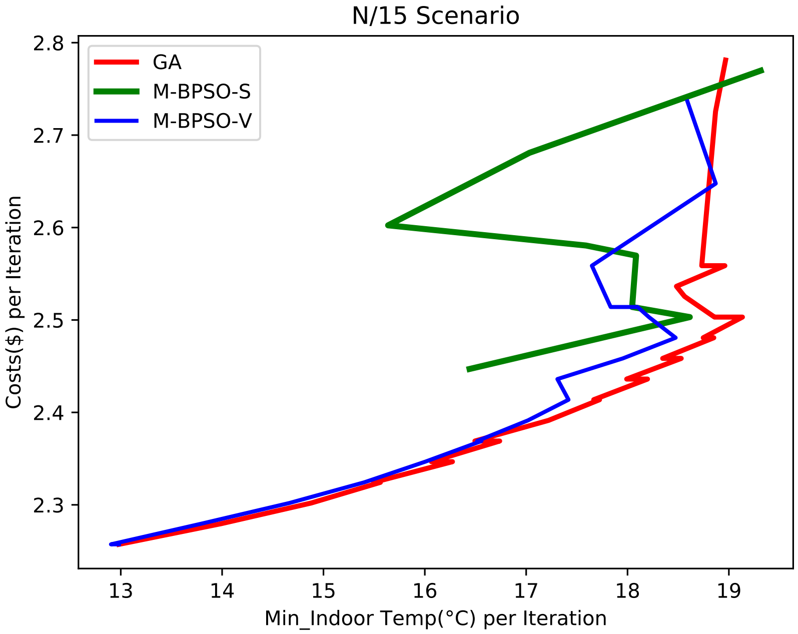


Figure A.6: Search space trajectory (N/15)

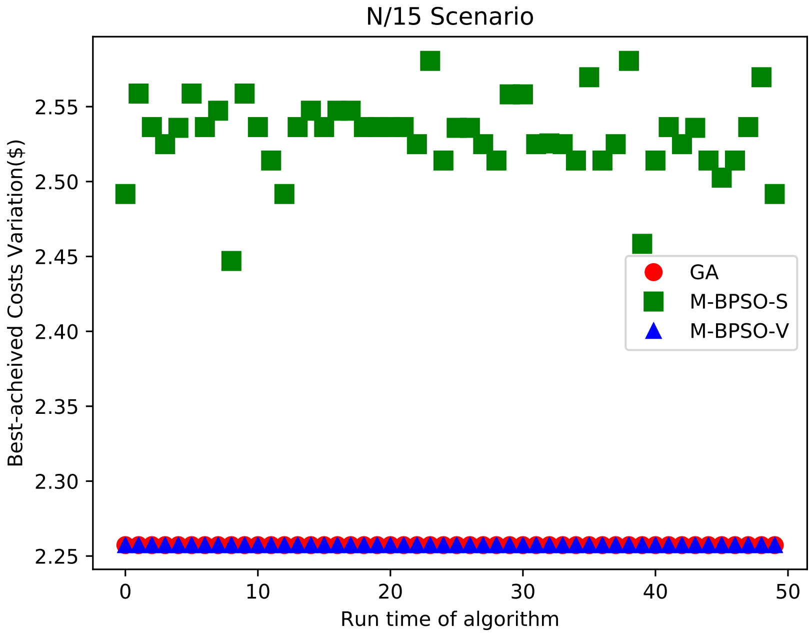


Figure A.7: The best achieved costs during each run (N/15)

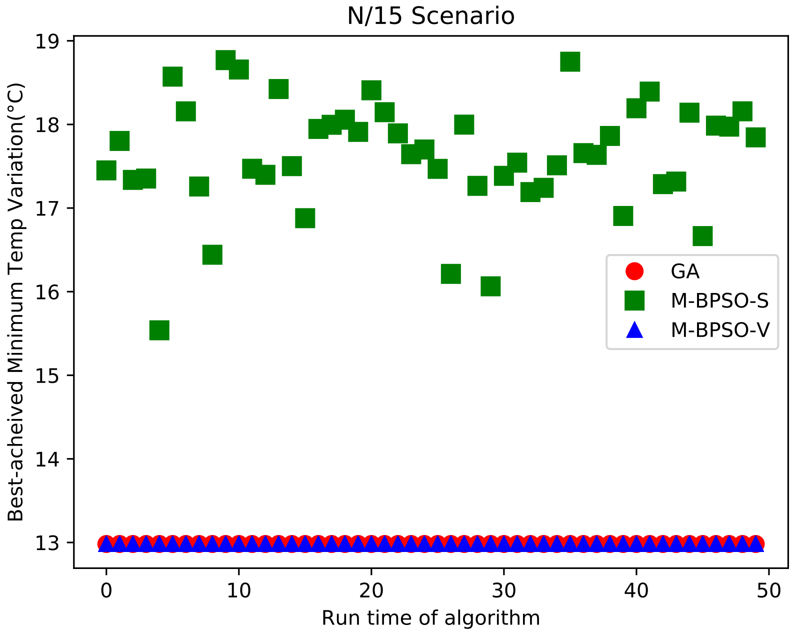


Figure A.8: The best achieved min temp during each run (N/15)

A.3 C/S/30 Scenario

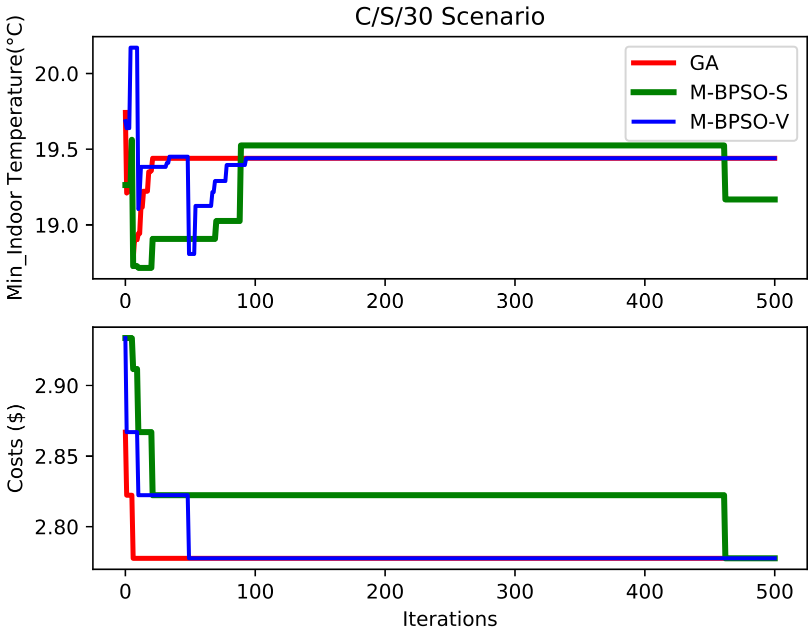


Figure A.9: Minimum indoor temperature and cost variation (C/S/30)

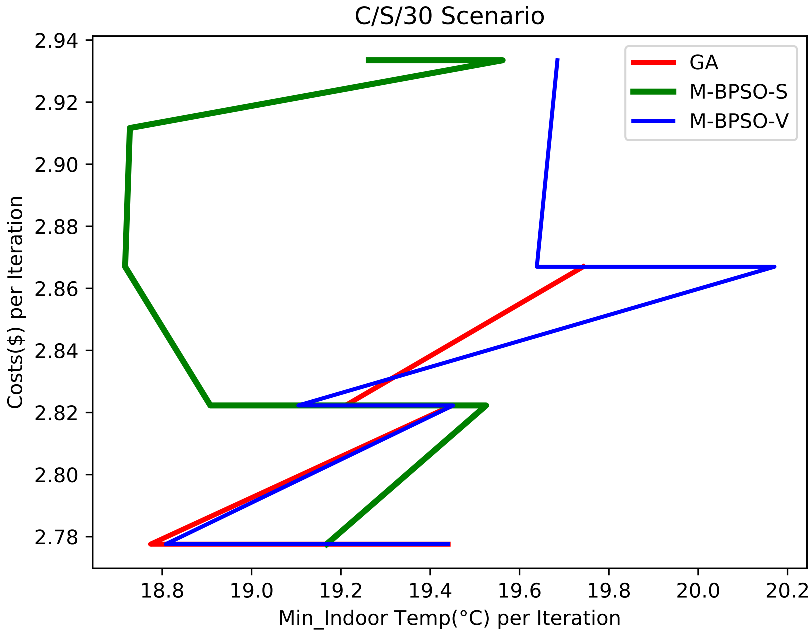


Figure A.10: Search space trajectory (C/S/30)

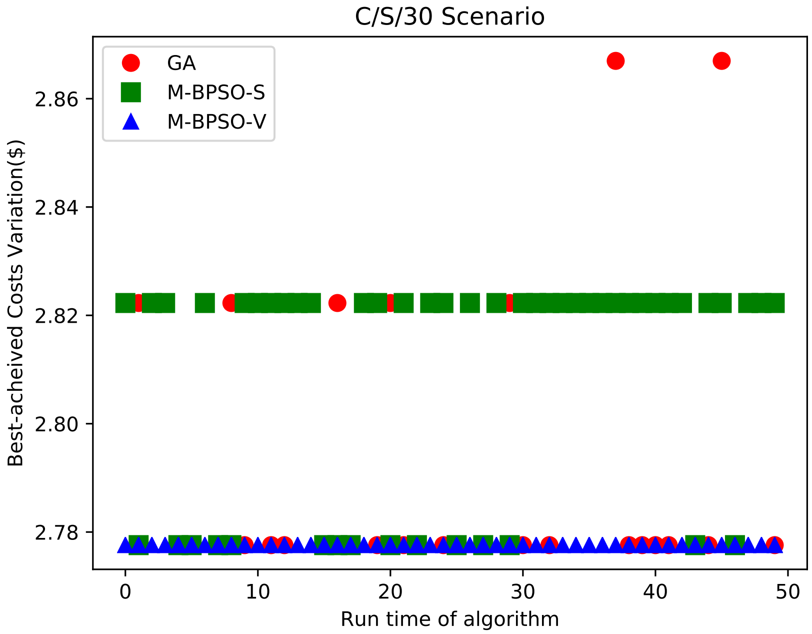


Figure A.11: The best achieved costs during each run (C/S/30)

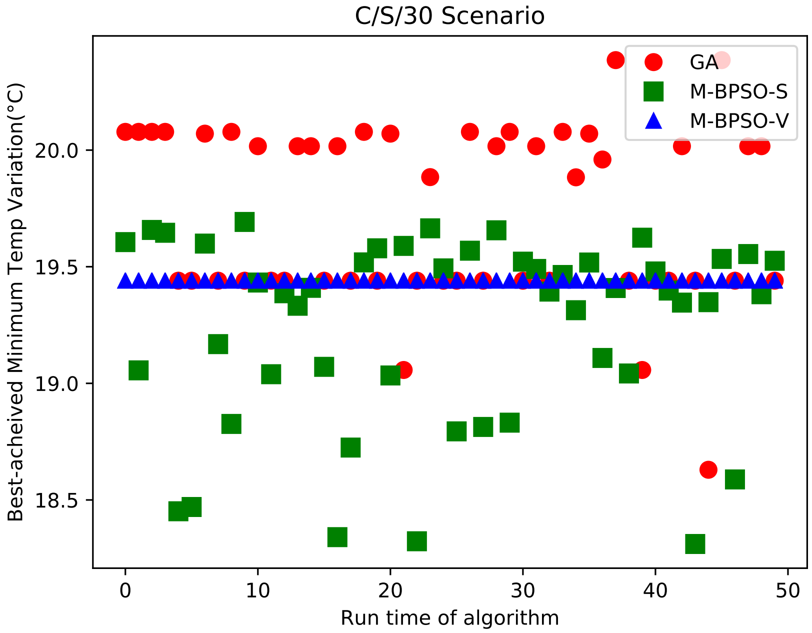


Figure A.12: The best achieved min temp during each run (C/S/30)

A.4 C/S/15 Scenario

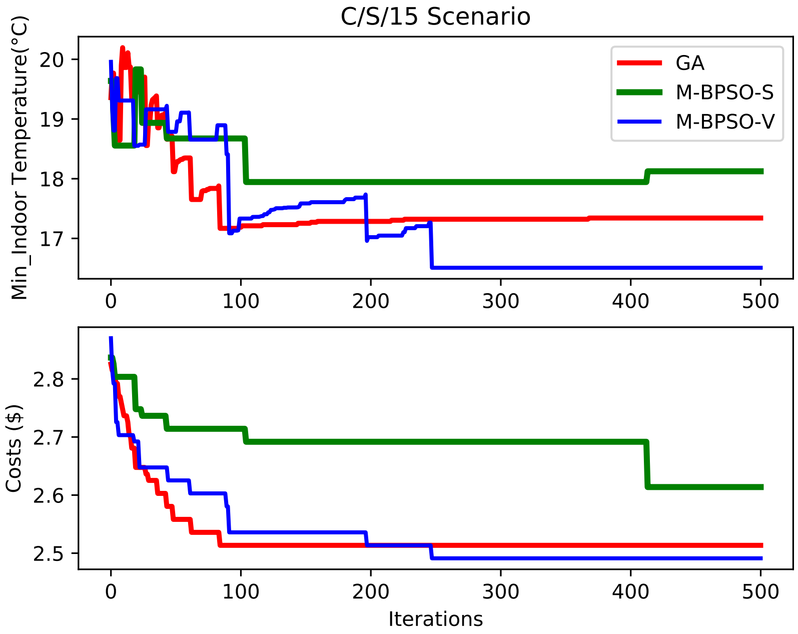


Figure A.13: Minimum indoor temperature and cost variation (C/S/15)

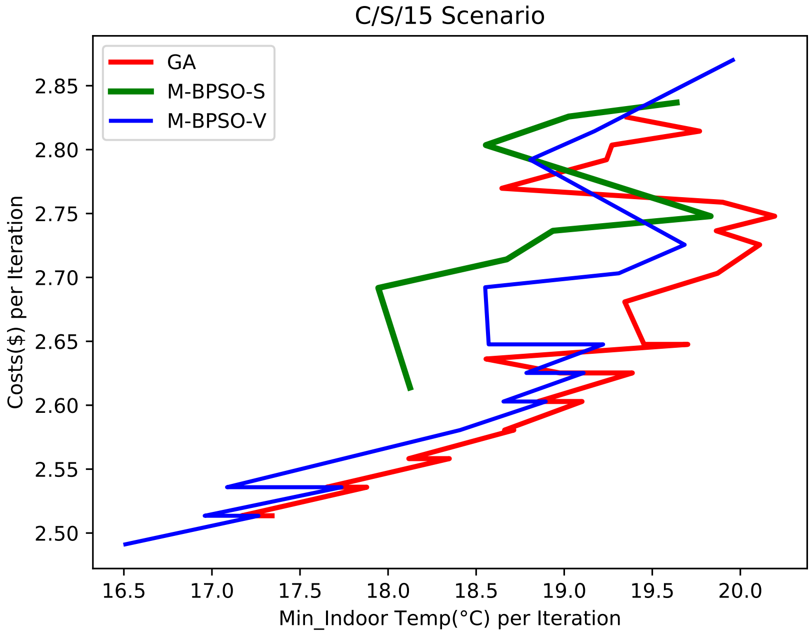


Figure A.14: Search space trajectory (C/S/15)

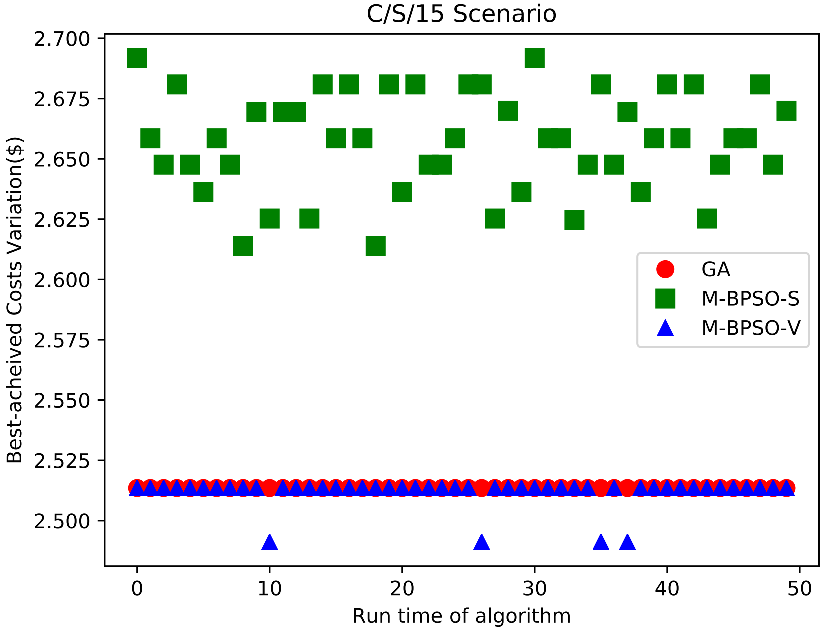


Figure A.15: The best achieved costs during each run (C/S/15)

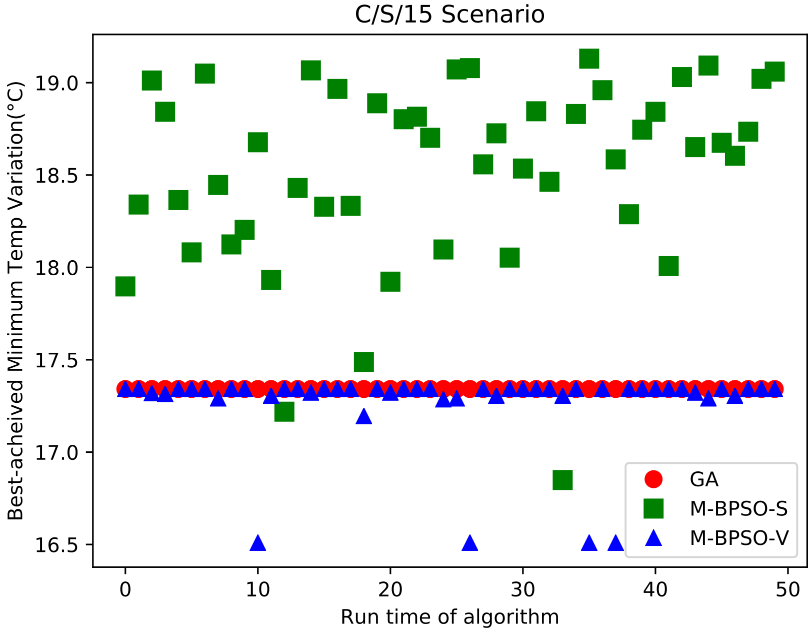


Figure A.16: The best achieved min temp during each run (C/S/15)

A.5 C/D/30 Scenario

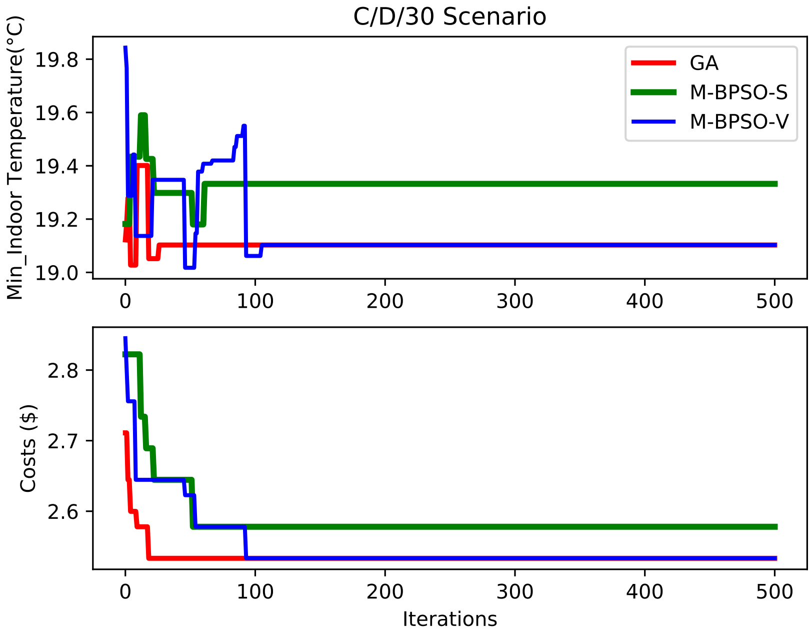


Figure A.17: Minimum indoor temperature and cost variation (C/D/30)

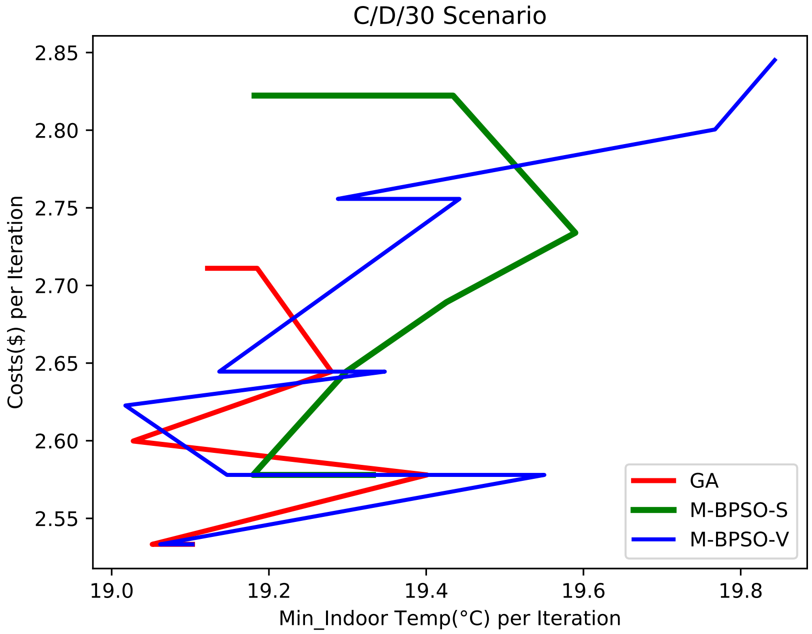


Figure A.18: Search space trajectory (C/D/30)

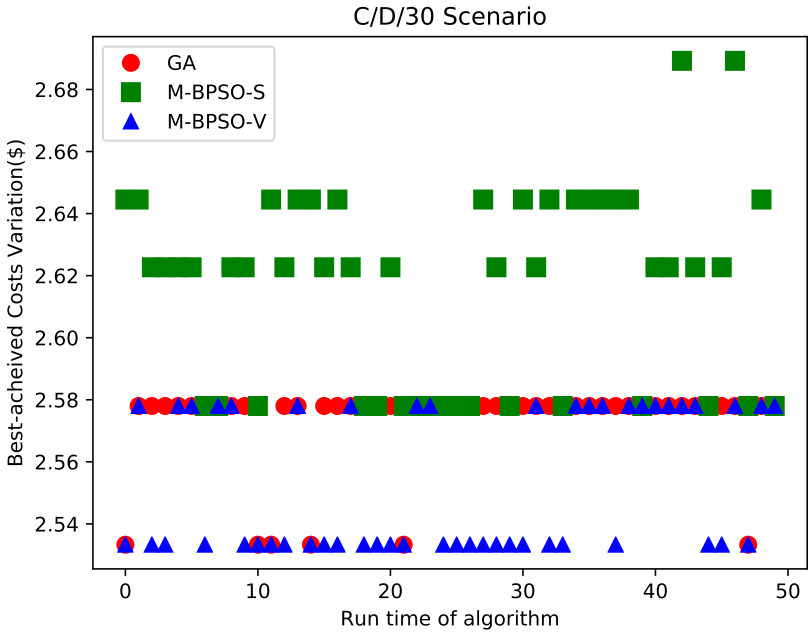


Figure A.19: The best achieved costs during each run (C/D/30)

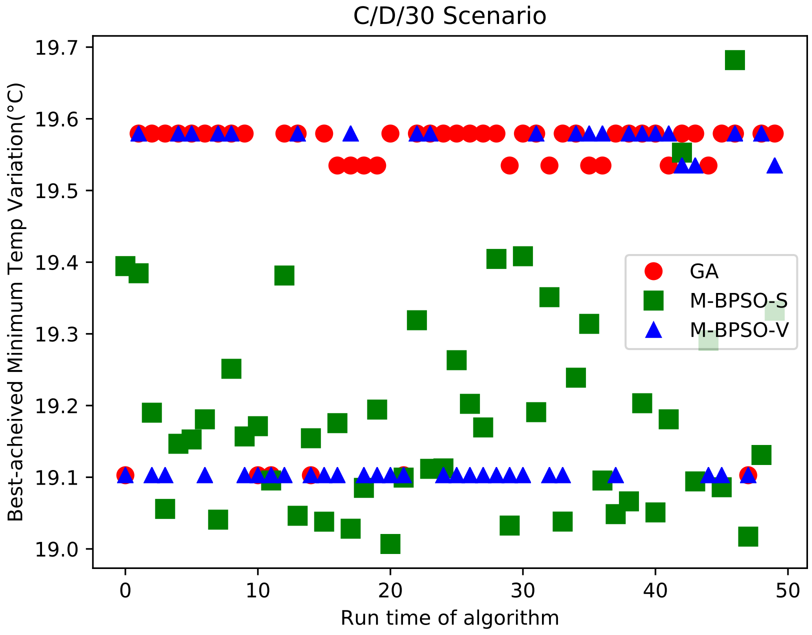


Figure A.20: The best achieved min temp during each run (C/D/30)

A.6 C/D/15 Scenario

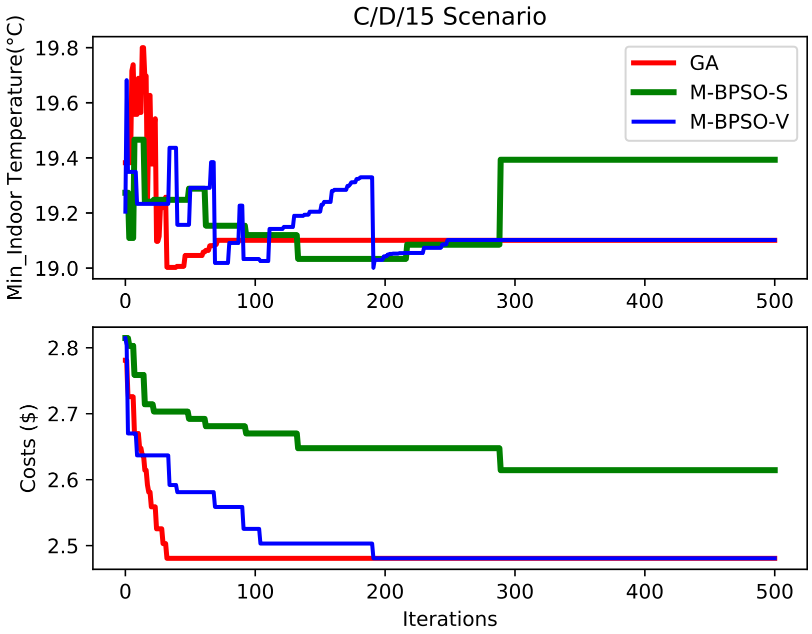


Figure A.21: Minimum indoor temperature and cost variation (C/D/15)

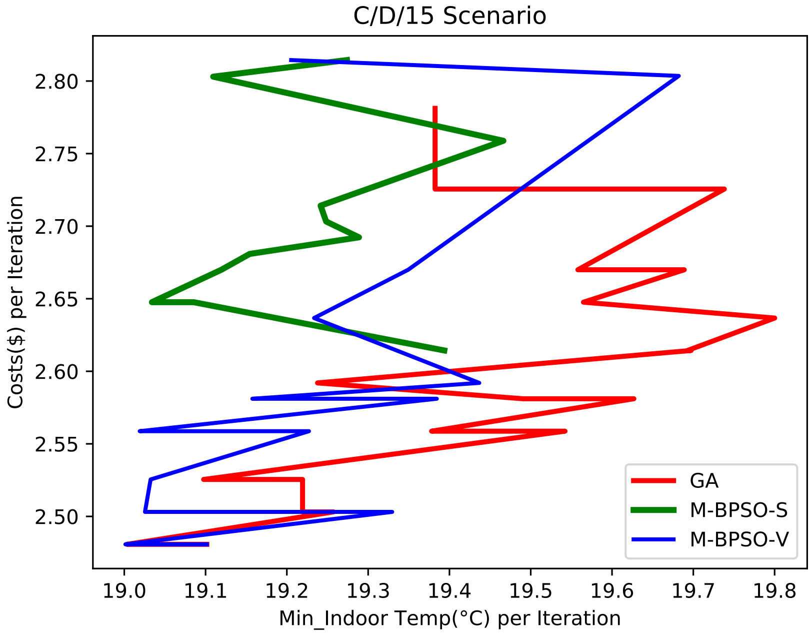


Figure A.22: Search space trajectory (C/D/15)

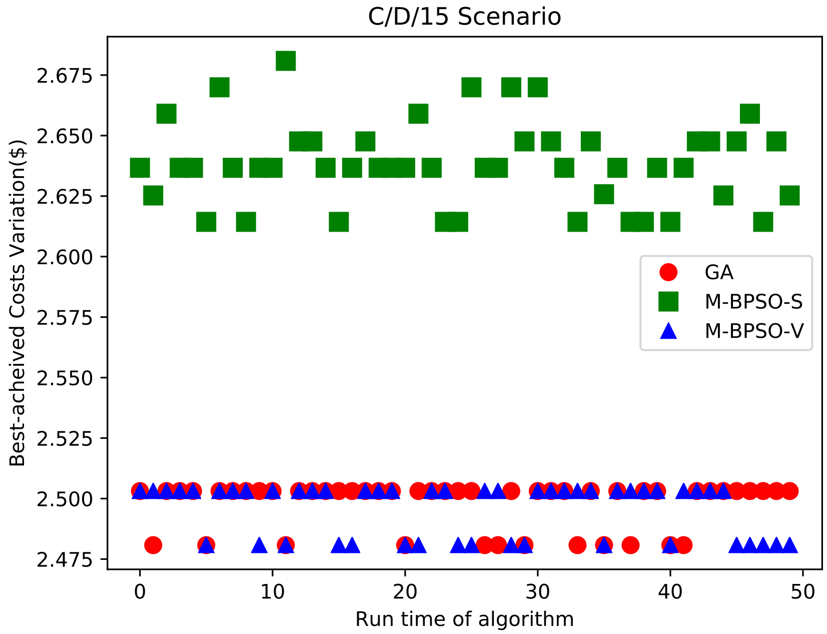


Figure A.23: The best achieved costs during each run (C/D/15)

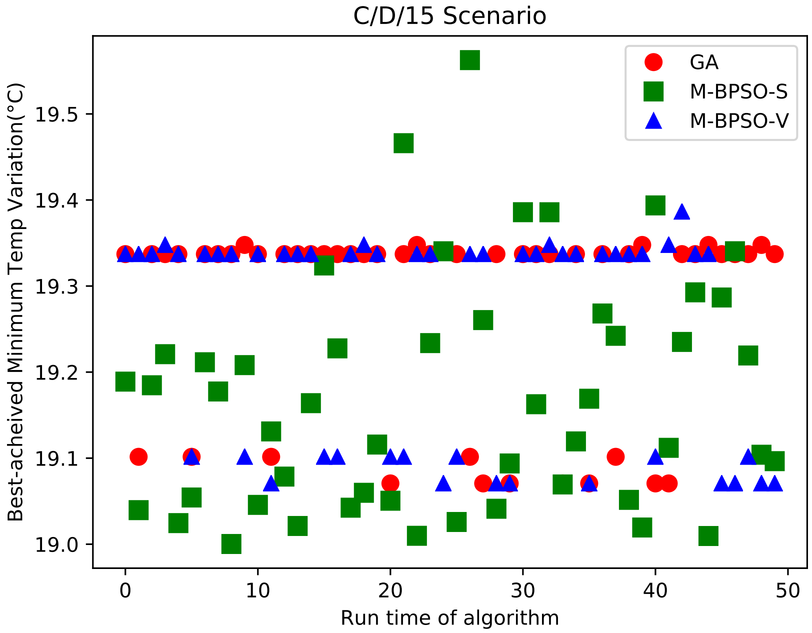


Figure A.24: The best achieved min temp during each run (C/D/15)