Knowledge-Based Systems 260 (2023) 110134

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

A heated stack based type-2 fuzzy multi-objective optimisation system for telecommunications capacity planning

Lewis Veryard [a],*, Hani Hagras [a], Anthony Conway [b], Gilbert Owusu [b]

[a] The Computational Intelligence Centre, School of Computer Science and Electronic Engineering, University of Essex, Colchester, UK
[b] British Telecom (BT) Labs, BT plc, Adastral Park, Ipswich, UK

ARTICLE INFO

Article history:
Received 18 March 2022
Received in revised form 10 November 2022
Accepted 14 November 2022
Available online 30 November 2022

Keywords:
Heated Stack
Optimisation
Fuzzy logic
Capacity Planning
Telecommunications

ABSTRACT

In this paper, we present the Heated Stack Algorithm (HS) which is a population based multi-objective evolutionary algorithm with temperature based on type-2 fuzzy logic meta-heuristic. Temperature plays a vital role in HS being used for two distinct procedures; Sorting and Crossover. In sorting, temperature is combined with the niche distance to determine the rank order of a population front. In crossover, the temperature of two population members are compared to determine the quantity of information to take from each parent. HS is a new optimisation algorithm capable of solving constrained real-world problems. This paper will present the HS application to a real-world capacity planning problem involving networking infrastructure. To proof the algorithm applicability to wider set of problems, we will report the HS results over a subset of the constrained multi objective problems used for optimisation competitions by the IEEE Congress on Evolutionary Computation (CEEC). In these problems we have compared to the popular NSGA-II and its successor NSGA-III. By use of the hyper-volume indicator, we find that the HS outperforms NSGA-II in 84% of cases, and outperforms NSGA-III in 69% of the cases.

1. Introduction

The internet, data and interconnectivity have become a pillar of modern society, being used in every part of people's lives, be this as part of their personal or professional lives. People require a reliable and fast internet connection in as many locations as possible. In the United Kingdom this connection is provided by a network that is broken down into two divisions; the access network connecting business and residences to an exchange, and the core network connecting all the exchanges together and to the wider world. The access network comprises of many different technologies such as copper cable, fibre optic cable, 5G, satellites and can be expanded to include any new technologies. The core network comprises data exchanges and heavy duty fibre optic cabling providing connections between exchanges and to the rest of the world. An exchange comprises bandwidth, cooling and power equipment. Bandwidth equipment is used to route data packets to their correct locations. The cooling equipment is used in order to keep the bandwidth equipment running in an optimal environmental condition. The power equipment is used to provide the necessary energy for the bandwidth and cooling equipment. As the ever-present demand for a faster network connection increases networking infrastructure in these exchanges must be upgraded in order to keep up [1], this upgrading of an exchange is known as ***capacity planning or capacity management***.

Upgrading each of the three parts of the exchange comes with its own challenges and constraints, but one overarching constraint applies to all three parts of the exchange where there must be a minimal loss of service. By far the simplest but most expensive part of the exchange to upgrade is the power equipment, as it underpins the other two and cannot be upgraded without disabling parts or all of the power within the exchange. Increasing the capacity of cooling equipment is more complicated to upgrade than the power equipment, but would still require the shutdown of bandwidth equipment. Hence, power and cooling equipment violate the overarching constraint to varying degrees, which only leaves the upgrade of bandwidth equipment. Thankfully newer bandwidth equipment is more power and temperature efficient than their predecessors, so we can decrease the cooling and power requirement by upgrading the bandwidth equipment whilst simultaneously increasing bandwidth. Unfortunately, bandwidth equipment is the most complicated and constrained environment to manipulate and upgrade.

There are varying types and structures of the bandwidth equipment each of which has its own set of requirements and constraints. For the basis of the explanation, we shall start with how the ideal exchange looks and then show how variations upon an exchange's structure can increase the complexity of the

* Corresponding author.
E-mail address: lhfver@essex.ac.uk (L. Veryard).

https://doi.org/10.1016/j.knosys.2022.110134

optimisation problem. First that all bandwidth equipment fits within a tree structure. Secondly the digital or software capacity management has been pre-optimised, and finally all equipment can be moved and is not under some special restrictions. In our simplified environment all bandwidth hardware comes under one of three categories; racks, cards and ports. Each of the three categories has a one to many relationship with the next categories (i.e. a card has many ports and a rack has many cards). A rack is used to supply physical mounting points for equipment whilst also providing cooling and power. A card directs packets from ports to their destinations by consuming power and producing heat. A port is a connection point for a cable be it fibre or copper. In order to upgrade a card all the ports must be moved to appropriate locations on other cards, and in order to upgrade a rack all cards must be cleared of ports and removed. Each rack, card and port has its own set of business and technical requirements and constraints including; hardware and software compatibles, power and cooling requirements, bandwidth usages/constrains, and configuration constraints. Another way of looking at capacity planning in telecommunications can see it as a constrained version of the theoretical computer science problems; the bin-packing [2] or the knapsack problems [3]. In which both problems require the sorting and arrangement of a set of objects into one or more containers across multiple layers.

In a national network there can be thousands of exchanges each built to suit the needs of its local area and upgraded to suit those needs over time. In this paper, we will be using one of these exchanges. However, in some locations, the structure and layout of a given exchange can be more complex. These complex structures of racks, cards and ports can have anywhere between three to nine layers and do not always follow the one to many relationships. Every port that is in use has a digital capacity architecture on it that has a relationship to other ports in the exchange, which must first be organised and considered when making physical decisions. Ports, cards and racks may have many constraints put upon each of them, reducing the possible valid state the entire exchange could exist in. In some cases, some ports and cards have special rules that must be interpreted by a human as they could be related to government, financial, military or international assets. In general, each of the exchanges follow the same trend as laid out, with an exchange being comprised of a set of racks, cards and ports, which need to be moved in order to make upgrades to the exchange.

The telecommunication data used for this problem has been supplied by British Telecom (BT) from their databases and is data from an exchange and is as such a digital representation of the real networking infrastructure within a building. In order to use this data, the relevant information needs to be extracted from the data and formatted so that it can be accepted by the HS and the other optimisation algorithms. After the extraction and formatting, the data is represented in two files one for the ports and cables and another for the cards and rack that make up the static infrastructure. In the ports and cables data there is a little over 1100 rows of data with each row representing a single port and this a single decision variable in the optimisation problem. Each row of the ports and cable data has 7 data fields including: the Port ID, its location on a card, its parent card, is the port occupied with a cable, the service type on that cable, the cable group id (if applicable) and bandwidth utilisation. In the cards and rack data file there is roughly 130 rows of data with each row representing a card with its associated information. Each row of the cards and rack data file has 5 data fields including: Card ID, its location on the rack, its parent Rack, the compatible service types and the bandwidth capacity. With an understanding of the data fields, the objectives and constraints of capacity planning problem used in this paper can be outlined. There are two objectives, first to remove all the ports from a given card, second to move the minimum number of ports from their original locations during this process. There are three constraints used in the telecommunication problem presented in this problem. First, some groups of cables must be kept together on the same card. Second is a given cable and its services compatible with the card it has been place on. Finally does the card breach its bandwidth capacity with the combined utilisation of the cables. Given that in this problem, there are a high number of specific edge case constraints that may only occur once in an exchange, we have simplified and generalised the constraints for the purpose of this paper, but the constraints presented are still valid and used as part of the optimisation process.

The terms capacity management or capacity planning has been used in many different problems in many different domains each with their own solutions. In manufacturing an agent based system has been used in [4], statistical modelling is used within virtual machine deployment in [5]. Supply chains across many different manufacturing processes use capacity management; system dynamics has been found to be effective in closed loop supply chains [6], and multi-objective optimisation being used in supply chain capacity management [7]. Capacity management is a term widely used for many different problems, but they all have some similarities; such as requiring the balancing of several systems or metrics in order to produce the most effective result or results. This broad idea of balancing several systems or metrics can also be applied to networking infrastructure including optical networks [8,9]. In telecommunications there are several different problems that are all labelled as capacity planning all of which rightly fit under this umbrella term. In one case the term capacity planning is being used to describe stock management within an exchange, allowing for the newest equipment to be brought ready for installation when it is most needed [10]. In another case the term capacity management is used when making investment decisions within a telecommunications network ensuring that new equipment can be installed in a timely manner [11]. In a final case the term capacity planning is used to refer to the modelling and decision making process used to increase the capacity of a network location, with the installation of new equipment [12]. All of these problems in both telecoms and the wider scope are in some sense a multi-objective optimisation or a multi-objective modelling problem and as such could have any number of solutions, but none of the solutions to these other problems are appropriate for our network capacity planning. These methods are all very good at their specific problems and domains, but none of these methods can adequately represent our problem space let alone fulfil all of the requirements and constraints of the capacity planning problem presented within this paper. The solutions to our capacity planning problem have a lasting real-world effect on nation's network infrastructure and incur high monetary costs to implement. Therefore, for these two reasons the Heated Stack Algorithm (HS) has been designed with this type of problem in mind.

The main contribution of this paper is the introduction of a novel multi objective multi constrained evolutionary algorithm called the Heated stack, which is capable of addressing short comings of well-known multi objective multi constrained algorithm including NSGA-II and NSGA-III in its ability to be able to balance fairly the exploration and exploitation of the algorithm. This has been realised by a new parameter called temperature which works by adding priority to newer members of the population in the selection and population reduction whilst also effecting the crossover of population members. This added priority reduces over generations and the extent to which priority is added reduces as the number of generations increase, but the effect of this priority can be increased as it is directly tied to the number of new population member that are created. This algorithm shows its own benefit where is has been used for a benchmark competition problems where it has been compared against NSGA-II and NSGA-III. When using the hypervolume indicator in the constrained multi objective problems (CMOP) HS can deliver a better solution front 84% of the time when compared to NSGA-II and 69% of the time when compared to NSGA-III. Not to mention that in the constrained multi objective multi chromosome problem space of capacity planning for telecommunications (the subject of this paper) HS will outperform NSGA-II 100% of the time and NSGA-III 64% of the time. This algorithm is highly needed for real world problems such as the telecommunication capacity planning which are characterised by a highly complicated decisions space with a high number of constraints, which cannot be effectively solved by NSGA-II and NSGA-III because they lack the ability to effectively explore the problem space prior to exploiting it. In this paper we were able to show that HS was able to solve the given problem and achieve results that were not achievable by NSGA-II and NSGA-III.

In this paper, we will present a novel constraint handling multi-objective evolutionary optimisation algorithm that uses an interval type-2 fuzzy logic systems for capacity planning within telecoms networks, which is also a capable of handling general optimisation problems. In Section 2, we will give an overview on simulated annealing and genetic algorithms. In Section 3 we will give a brief overview of fuzzy systems. In Section 4 we will present the proposed HS for solving capacity planning in telecoms and constrained multi objective problems. In Section 5 we will present the experiments and results to the capacity planning problem and a sub-set of the constrained multi objective problems used for optimisation competitions by the IEEE Congress on Evolutionary Computation (CEEC). Finally in Section 6 we conclude the paper and outline our future work.

2. An overview on simulated annealing and genetic algorithms

Optimisation is a well-developed field of computational intelligence with problems being broken down into several categories; single objective, multi-objective, and constrained multi-objective problems. In optimisation a set of decision variables {x1…xn} are manipulated in order to satisfy a set of one or more objectives f (O1…On). The area a decision variable can be manipulated across is known as the decision space. Whereas the areas that an objective can exist across is known as the objective space.

2.1. Single objective optimisation

A single objective optimisation problem consists of one objective that usually requires either maximising or minimising. There are many types of evolutionary single objective optimisation including; Particle Swarm [13], Big Bang Big Crunch [14] and Ant Colony Optimisation [15]. Genetic Algorithms (GA) [16] and Simulated Annealing (SA) [17] have both had great success in the field of optimisation.

2.1.1. Genetic algorithm

The GA takes the main principles of Darwinian evolution with mating, generations and survival of the fittest. A GA is made up of a population defined as a set of chromosomes, a chromosome is a collection of genes with each gene representing one of the decision variables within a specific problem. The algorithm goes through each of the operations every generation or loop, these are as follows; selection, crossover, mutations and reduction. The selection operator is used to determine which member of the population should crossover or produce off-spring, there are a number of different techniques with the most popular being tournament or roulette wheel. Once the population has been split and sorted into two groups by the selection operator the chromosomes are picked two at a time for crossover. The crossover operator is used to add new individuals to the population by combining two current members together. The original members of the population are known as parents and the newly created members are known as children.

Once children have been created there is a random chance that they can be mutated. Just like in nature, mutation adds changes to one or more genes, which in-turn injects some genetic diversity into the population. Once children have had a chance to be mutated they are then added back into the population. After all parents have been mated, the population will now be double the size that it was before crossover began. The reduction operator (sometimes called elitism) is used to remove half of the population. Keeping in form with survival of the fittest the lowest ranked members of the population are removed. Members of the population are ranked based on how well a chromosome fulfils some objective criteria.

2.1.2. Simulated annealing

Simulated annealing is based on the idea of statistically modelling the annealing process of solids such as iron in smiting or silicon in semiconductor production. The roots of the algorithm come from the metropolis algorithm [18] using similar ideas. The idea of annealing is to heat or introduce energy into a material so it can be shaped. When the system has a high heat, it is easy to make large changes to, but as the system cools it becomes harder to make larger changes so only smaller changes are made, this continues until no new changes can be made without the addition of heat. Simulated annealing takes this principle of changing heat states and uses it as an optimisation technique. It represents a single potential solution of decision variables and manipulates it based upon the temperature within the system.

A solution changes over time with the use of two operators, the first is used to present potential changes to the original solution by manipulating the decision variables and storing it as a new solution. The second is used to determine whether or not to accept the changes presented and change the original solution. This acceptance is calculated by evaluating the new solution and calculating the acceptance metric which incorporates the temperature within the algorithm. This acceptance metric is based upon the Boltzmann probability factor [19]. If the new solution is accepted it becomes the original solution and the temperature is reduced and the stopping criteria is checked. The two operators are invoked iteratively until the stopping criteria is met. Simulated annealing starts with a high temperature and reduces the temperature by the cooling rate every cycle, the cooling rate is a number between but not including 0–1. The stopping criteria is a small user defined value, typically a value smaller than one, which stops the algorithm once the temperature is below it.

2.2. Multi objective optimisation

Multi objective optimisation is very similar in principle to single objective optimisation where a set of decision variables represent a solution and are manipulated to achieve a desirable result. The only difference is that there is now a set of Objectives f $(O_1 \ldots O_n)$ instead of a single objective. There are several successful multi objective optimisation algorithms including; cARMOEA [20], AnD [21], CCMO [22], NSGA-II [23] and NSGA-III [24].

The complexity of a Multi objective optimisation problem dwarfs that of the single objective problem space. The most prominent issue is how to determine if one solution is better than another. This problem is addressed with the use of dominance

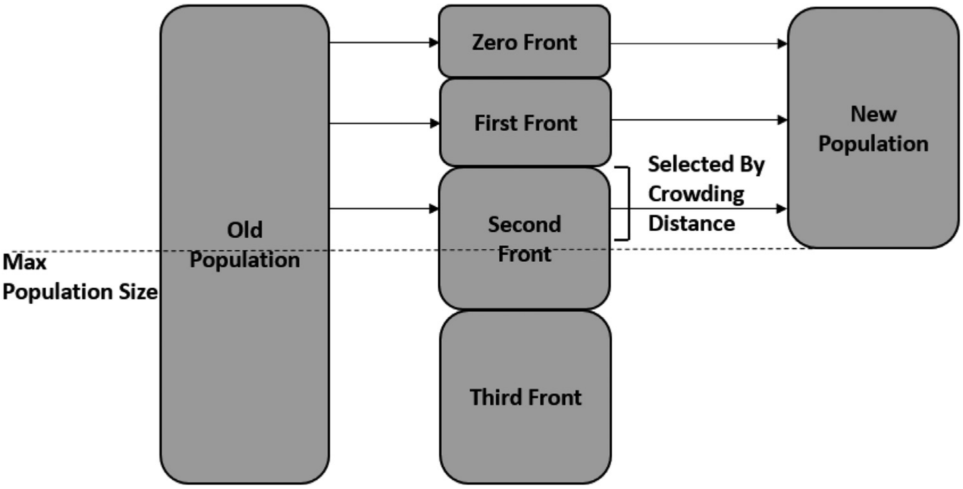

Fig. 1. NSGA-II population reduction sorting.

rules, and the use of fronts. Most multi objective evolutionary algorithms (MOEA) use both dominance rules and a Pareto front [25] in an attempt to sort a set of solutions based upon conflicting objectives. In NSGA-II and NSGA-III the dominance rules are used as part of the population sorting process. Given there are two solutions A and B that both have more than one objective. A is dominant over B if all of A's objectives are no worse performing than B's objectives and at least one of A's objectives is better performing than one of B's objectives, otherwise A is not dominant over B [25].

In NSGA-II and NSGA-III, once every member of the population has had its dominance checked the population is sorted into its corresponding front. The worst performing members of the population are in the worst or last front, and the best performing members are in the zeroth front. For every problem there is theoretically a best front known as the Pareto front, which contains a set of Pareto optimal solutions. The Pareto optimal [26] solution is one that cannot be dominated over by another solution that have been found or will ever be found, allowing it to represent one of the best solutions.

2.2.1. NSGA-II

NSGA-II or Non-Dominant Sorting Genetic Algorithm [23] is wildly known in the literature to have high quality results in multi objective optimisation problems [27]. It has similar functionality to a single objective GA with; selection, crossover, mutation and reduction all being very similar. The major difference is whenever the population requires sorting, the population goes through a two stage sorting process. In the first stage the population is sorted into fronts using the domination rules. The second stage is a density metric named crowding distance which calculates how close given solutions neighbours are on a front. Crowding distance is measured using Euclidean distance as shown in Eq. (1), where p_1 and q_1 are objective values from one solution and p_2 and q_2 are objective values from a second solution.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 + q_2)^2} \tag{1}$$

A set of solutions are sorted using this process for one of two operators; selection and reduction. During selection the population is split randomly into two sub sets and sorted to allow the best solution from one subset to mate with the best from the other subset. To reduce the overall population size by half after crossover, the whole population is sorted and then the worst performing subset are removed from the population as shown in Fig. 1.

2.2.2. NSGA-III

Non-Dominant Sorting Genetic Algorithm III (NSGA-III) is an updated version of NSGA-II which follows a similar trend of a GA using the four operators: selection, crossover, mutation and reduction. In a similar fashion to NSGA-II it uses the domination rules [25] for sorting, but it no longer uses Crowding Distance, instead the methodology of Niche-Preservation is applied to sort our population. The domination rules are used to remove the worst performing member for the population, in Fig. 1 a portion of the new population comes from the second front which in NSGA-II is sorted by crowding distance, in NSGA-III the Niche-Preservation operation is used. The Niche-Preservation sorting method can be broken down into three stages; Normalise, Associate, and Niche-Preservation. In order to normalise we must calculate the min and max values in the objective functions for each objective. First the ideal point of each objective function is calculated, this is usually the minimum possible value. Second the maximum recoded point across all generations is stored for each objective. The Min value is determined by the smallest number possible for an objective, whereas the Max value is determined by the highest number seen across all generations. These two values are used as our Min and Max values in the Normalisation operation of the objective values as shown in Eq. (2).

$$norm(fO_i) = \frac{Max - fO_i}{Max - Min} \tag{2}$$

Each of the chromosomes in the front now has an N-dimensional coordinate associated with it based upon the normalised value of its objective function. These points can now be associated onto a predetermined reference plane. The reference plane can be created in a systematic manner or one can be supplied by a user. One suggested method is the Das and Dennis's [28] systematic approach that places points on a normalised hyper plane. Fig. 2 shows an example of a 3-Dimensional reference plane with 15 reference points, with the apex of these point being *f1(0,0,1)*, *f2(1,0,0)* and *f3(1,0,0)*. The association operator takes the normalised N-dimensional coordinates and places them on the reference plane. Once placed upon the plane they are associated with their closest reference point. If a front's population is uniformly spread out across the plane, each point will have a low number of chromosomes associated with it. Alternatively, if a front's population is not uniformly spread across the plane a few reference points will have the majority of the chromosomes associated with them. Once every member of the front has been placed upon the reference plane each reference point counts the number of associated chromosomes.

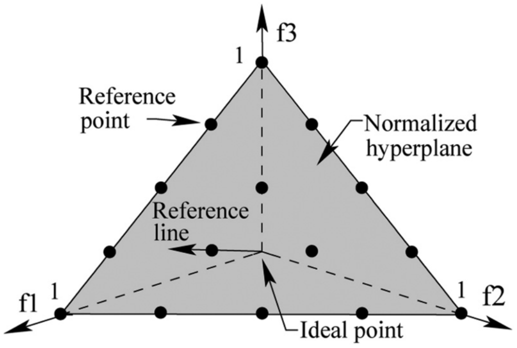

Fig. 2. Example of a three dimensional hyper plane with reference points [29].

For the Niche-Preservation operator two versions of the reference plane are used at the same time, the first plane P_1 has all the members of the population that are already in the new population (e.g. Zero front and First front in Fig. 1), the second plane P_2 are all the points from the next front (e.g. Third Front in Fig. 1). The points in P_1 and P_2 are identical aside from having different associated chromosomes. All chromosomes on P_1 are considered to be part of the new population, and thus any new chromosomes added to P_1 are added to the new population. Points are removed from P_1 and P_2 when the niche count of points in P_2 are equal to zero. To begin, points in P_1 are ordered in ascending order by the number of chromosomes, then a chromosome from the P_2 from the same point is moved to P_1. If the Point on P_2 has more than one associated chromosome then the one with the lowest perpendicular distance is moved to P_1. If two chromosomes have the same perpendicular distance one is picked at random. If a point in P_2 no longer has any chromosomes associated with it, then the same point is removed from P_1. This process continues until the P_2 and thus the new population has the target number of members. The idea is that as members of P_2 are added to P_1 they are added from across the reference plane, therefore ensuring the population is more equally spaced across the search space.

2.3. Constraint handling in NSGA-II & NSGA-III

As the optimisation problems become more complex, they begin to incorporate constraints alongside objectives. In the presence of constraints, a solution can either be described as feasible where it is not in violation of its constraints or infeasible where it is in violation of its constraints. This means when comparing two members of a population there are three possible situations: First, both solutions are feasible; second, one solution is feasible and the other is infeasible; third, both solutions are infeasible. A solution is infeasible by the degree that it has violated the constraints, therefore two solutions can be infeasible to different degrees.

In order to handle the constraints NSGA-II [23] and NSGA-III [24,29] use an extended set of domination rules, where a new set of rules are applied before the original dominance rules. Given there are two Solutions A and B and given that both solutions have a quantifiable objective score and constraint violation value, these rules can determine dominance and split a population into fronts. A solution is feasible if it does not violate any constraints, whereas a solution is infeasible if it violates one or more constraints. Solution A constraint dominates Solution B if A is feasible and B is not. Solution A constraint dominates Solution B if A and B are both infeasible but A has a smaller constraint violation than B. Solution A constraint dominates B if both solutions are feasible considering that A dominates B using the domination method described in Section 2.2.1.

Let us consider an example, given that there is a problem that consists of two real number variables $(V_{1,}V_{2)}$ which is trying to achieve two objectives (O_1, O_2):
O_1: MINIMISE: V1 + V2
O_2: MAXIMISE: V1 * V2

Given that these objectives exist, three constraints (C_1, C2, C3) can be introduced to constrict the problem search:
C_1: V_1 must be a prime number
C_2: V_2 must be a prime number
C_3: $10 < V_1, V_2 > 20$

Given the set of objectives and constraints the following example solutions will either be feasible or infeasible and thus can be ranked.

Solution 1		
V_1: 9	O_1: 47	C_1: Violate
V_2: 38	O_2: 342	C_2: Violate
		C_3: Violate
Solution 2		
V_1: 11	O_1: 30	C_1: Satisfy
V_2: 19	O_2: 209	C_2: Satisfy
		C_3: Satisfy
Solution 3		
V_1: 13	O_1: 29	C_1: Satisfy
V_2: 16	O_2: 208	C_2: Violate
		C_3: Satisfy

Each of these three solutions can be determined as feasible or infeasible and given an infeasibility score. Solution 1 is infeasible as it violates all the constraints which gives it an infeasibility score of 3. Solution 2 is feasible as it satisfies all the constraints and therefore has an infeasibility score of 0. Solution 3 is infeasible given that it violates C_2 giving it an infeasibility score of 1. Members of the population can be ranked from best to worst based upon their infeasibility scores. Solution 2 is the best ranked with an infeasibility score of 0, then solution 3 with an infeasibility score of 1 and finally solution 1 with an infeasibility score of 3.

2.4. The multi-arm bandit problem & exploration vs. exploitation

A problem can be tackled from many different approaches, and it is important that each problem is solved with the correct type of approach. In machine learning there is this idea of the Multi-Armed Bandit Problem [30,31] where there is only a limited amount of resources to solve a complex problem. The classic example is the gambler in front of a set of slot machines where they have a limited amount of money and time to generate the most money out of said slot machines. This poses the question of how does the gambler maximise the reward from their money and time, do they only try with one slot machine, or do they change to another after some time interval, or try a different strategy entirely. By abstracting this idea it can be applied to optimisation problems. Where there is a limited amount of time to solve a problem and there is no one approach that will provide the best solution, so the question becomes how does an optimisation algorithm best spend its time searching the solution space to best achieve its goal.

In optimisation there is this idea of Exploration vs. Exploitation. Exploration is the act of looking for new unseen solutions, these solutions do not necessarily immediately improve upon the solution space, but could lead to better unseen solutions. Exploitation is the act of trying to get the most out of the solutions and getting a better solution given what already exists within the solution space. In NSGA-II the domination ranking system is used to exploit the best solutions and ensure the best in the population survive until the next iteration. In order to explore and keep some

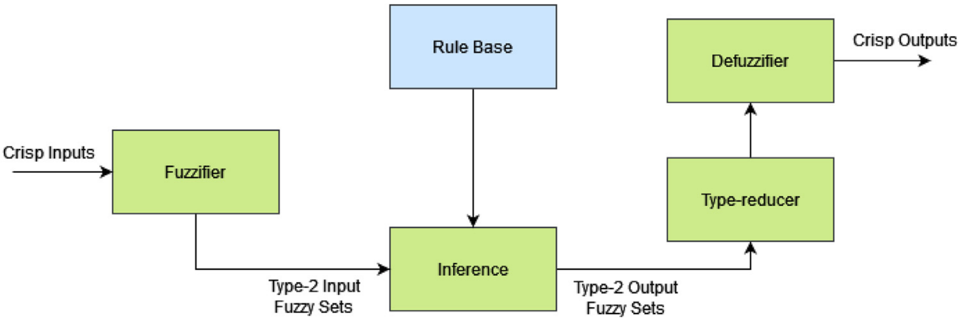

Fig. 3. Type-2 fuzzy logic system [34].

diversity in the population, NSGA-II uses the mutation operator whilst also using crowding distance indicator in an attempt to spread solutions across the front [23]. NSGA-III uses the same methodology of domination to exploit the population during crossover and population reduction with some minor differences. In order to explore and increase population diversity it uses the mutation operator and the niche distance method.

3. Brief overview of Type-2 fuzzy logic

Interval Type-2 Fuzzy Logic System (IT2FLS) builds upon a Type-1 Fuzzy Logic System which takes the idea of traditional logic and extends it so there are degrees of logic as opposed to being clear cut. In traditional logic something can exist in one of two state “True” or “False” but in fuzzy logic something can exist as both true and false to differing degrees at the same time [32]. This idea can be extended to more than just true and false, but to also more complex information that is traditionally only understood in human linguistic languages such as the temperature being warm or hot. Other examples include the speed of a vehicle being slow or fast and food being tasty or horrible. Fuzzy logic excels at capturing the imprecise nature of human language and applying it to the precision of a computer [33].

The Interval Type-2 Fuzzy Logic System (IT2FLS) is shown in Fig. 3 is comprised of 5 components: Fuzzifier, Rules, Inference Engine, Type Reducer and Defuzzifier.

The Fuzzifier takes a crisp or real number and transforms it into a Type-2 membership value. A Type-2 membership function is defined as a 3 dimensional area plotted on a domain axes with a Footprint of Uncertainty (FOU) [35]. In this paper we use an interval Type-2 fuzzy set to represent input and outputs as they are computationally more efficient than general Type-2 fuzzy sets. Fig. 4 shows an example of an input being fuzzified, the input of 12.5 has a membership within the shape with a membership degree of between 0.16 and 0.5.

Once inputs have been fuzzified they are passed onto the inference engine that activates rules using the input Type-2 fuzzy sets. The rule base works as a set of instructions to map the input sets to the output sets. In the final step the Type-2 output sets need to be returned into real crisp numbers, which can be achieved in one of two ways. Either they can be type reduced into Type-1 sets and then Defuzzified, or they can be directly defuzzified from a Type-2 set. There are several methods of type reduction and direct defuzzification [33].

In this paper we use the centre of sets type reduction as it has a reasonable computational complexity that lies between the expensive centroid type reduction and the simple height and modified height type reduction, which both have issues with one rule firing [35,36]. Once the output sets have been type-reduced they are then defuzzified by taking the average of their reduced values.

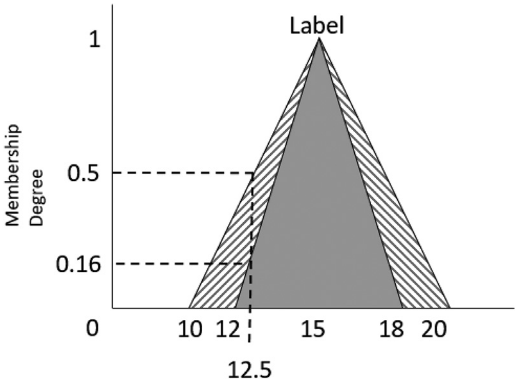

Fig. 4. Interval Type-2 fuzzy sets (a) primary membership domain.

Type-2 fuzzy logic systems been successfully applied in a large variety of domains including: unmanned aerial vehicles [37], explainable segmentation of trees [38], resilient routing in uncertain environments [39], computing with words [40], strategic telecoms network design [41], streaming data regression [42], financial investments [43] and electric vehicle breaking control [44].

4. The proposed heated stack based Type-2 fuzzy multi-objective optimisation system

The proposed HS takes its inspiration from evolutionary algorithms, specifically NSAG-II and NSAG-III and employs fuzzy logic within its sorting processes, using the Genetic Algorithms (GAs) ideas of selection, crossover, mutation and population control. Fig. 5 shows the flow of the system, which begins by producing an initial random population for the specific problem configuration. It proceeds to enter a loop where it goes through selection and crossover, with some of the children being randomly mutated. Once the selected members of the population have been crossed over, the entire population then undergoes being evaluated, sorted and reduced back down to a predefined population limit. Finally, each member of the population has their temperature reduced by a specific amount and the cycle starts the next loop, continuing for a set number of iterations. The following subsections details the proposed system.

4.1. Data structure

HS was originally designed for optimising multi-layer network representation problems [45]. Due to this original problem the HS has been designed with a multi-faceted solution able of representing a multi-layer solution. These solutions are stored in a

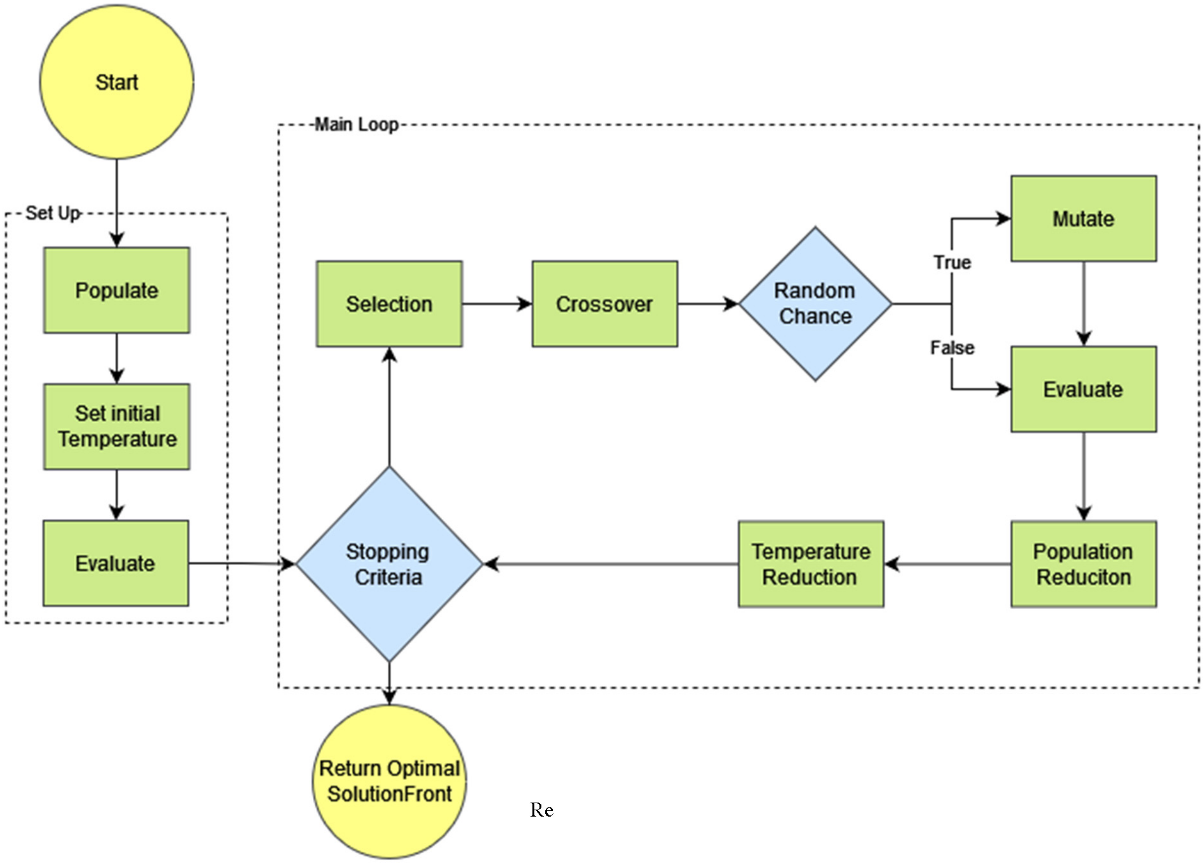

Fig. 5. A flow chart outlining the system flow of the heated stack algorithm.

Stack represented by a set of chromosomes C, with each chromosome representing a set of genes G, with each gene representing a decision variable. Eq. (3) shows the solution representation.

$$Stack = \{C_0\{G_{00} \ldots G_{0n}\} \ldots C_n\{G_{n0} \ldots G_{nn}\}\} \tag{3}$$

A Stack can represent a problem that requires multiple chromosomes or just one chromosome without impacting the algorithmic processes of the proposed HS. A stack has the additional variable temperature used within the sorting & crossover processes. The proposed HS uses a population of solutions to explore a problem space and uses a modified idea of survival of the fittest in order to sort the population of solutions. Sorting takes place during two methods; Population reduction and selection.

4.2. Population sorting

The Population is sorted for two reasons; population reduction and selection. In these two cases the population is sorted in almost the same fashion, with members of the population ranked based upon their domination first, then through the use of the IT2FLC for population sorting.

4.2.1. Population reduction

In population reduction, this sorting process happens in 2 stages; in the first stage members of the population are sorted into their front based upon the constraint domination rules as described in the earlier sections about domination and constraint handling in Evolutionary Algorithms. Then each front is added to the new population in turn only if the entire front can be added without exceeding the population limit, otherwise the final front is sorted and the best ranked members of the final front make it into the new population. Evolutionary Algorithms use the overarching idea of survival of the fittest where the best ranked solutions are kept within the population from generation to generation. In the proposed HS we take this idea of survival of the fittest and also consider the temperature of each population member. In the second stage, the ***temperature and Niche Distance of each member*** of the population is considered as inputs of an Interval Type-2 Fuzzy Logic System (IT2FLS). The temperature and score of the entire population is normalised between 0–1, allowing the fuzzy inputs as shown in Fig. 6, to be used no matter the problem. The min and maximum values for normalisation are updated from the population every generation. The min and max values represent a global min and max, meaning the min and max throughout every generation thus far.

Fig. 6a shows the input set for temperature, whereas Fig. 6b shows the niche distance which is calculated in the same manner as shown in NSGA-III, using an equidistance spread of reference points across the normalised hyper plane. Table 1 shows the rule base used for sorting and Fig. 7 shows the sorting Fuzzy sets. The rule base has been created with the intention of sorting by the niche distance but with the preference of population members with a higher temperature. Once all members of the front have been given a new score by the IT2FLS they are then sorted into ascending order. If this entire sorted front was added to the new population, the new population would exceed the population limit, so the new population count is brought up to the population limit by adding the best ranked members in turn.

4.2.2. Selection

The Selection process sorts the population for crossover, which in turn creates 2 new members of the population for every 2 parents, doubling the size of the population. In selection, the population is randomly divided into two groups, similarly to the

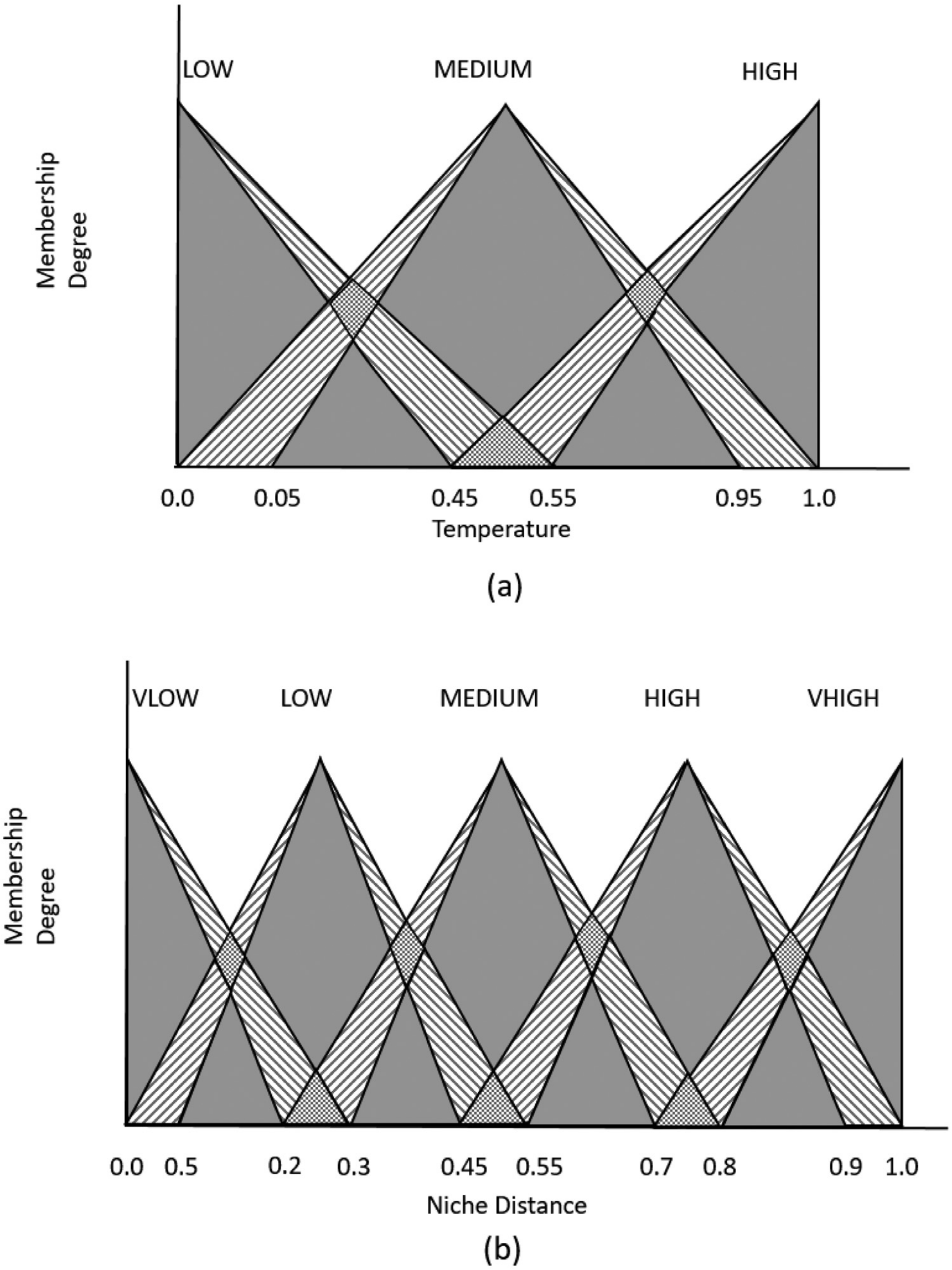

Fig. 6. (a) Input fuzzy set for temperature, (b) Input fuzzy set for niche distance.

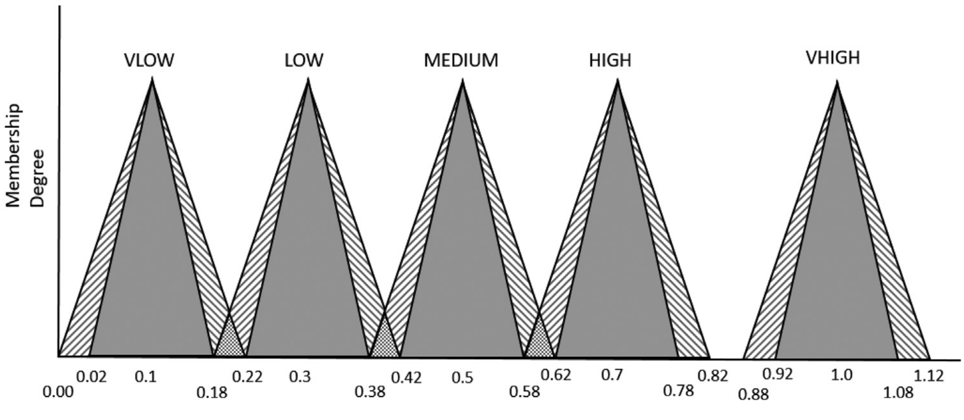

Fig. 7. Fuzzy sorting set.

Table 1
Rule base for fuzzy sorting.

Temperature	Niche distance	Sorting
Low	VLow	High
Low	Low	Med
Low	Medium	Low
Low	High	VLow
Low	VHigh	VLow
Medium	VLow	VHigh
Medium	Low	High
Medium	Medium	Medium
Medium	High	Low
Medium	VHigh	VLow
High	VLow	VHigh
High	Low	High
High	Medium	High
High	High	Med
High	VHigh	Low

Table 2
The fuzzy rule base for crossover quantity.

Chromosome one temperature	Chromosome two temperature	Crossover quantity
Low	Low	VLow
Low	Medium	Low
Low	High	Medium
Medium	Low	Low
Medium	Medium	Medium
Medium	High	High
High	Low	Med
High	Medium	High
High	High	VHigh

population reduction process these two groups are first sorted by their domination count, then their fuzzy score. This allows a balance of exploration and exploitation, where the best ranked member of the population does not always crossover with other best ranked members of the population.

4.3. Crossover

The crossover process also takes into account the temperature of the 2 parents when creating the children, with parents with a higher temperature passing on more information than parents with a lower temperature. This temperature comparison for crossover quantity is executed with the second IT2FLS. Fig. 8a shows the input set, Fig. 8b shows the output set for crossover quantity, Table 2 shows the rule base. Once again, all inputs are normalised between 0–1 based upon the entire population's global min and max values.

When new members of the population are created, they take decision variables or genes from both parents, the quantity of which is determined by the crossover quantity fuzzy logic system. Additionally the parents impart their temperature on to their children, with the children taking the temperature from the hottest parent and increasing it by a predefined value. Once a child solution is created there is a random chance that the Mutation operation is called, making several small changes to the new child member before it is added to the population. The mutation operation is a way of encouraging a diverse population of solutions by inputting random changes into some of the new solutions.

4.4. Temperature

Temperature is an integral part of HS being used for selection, population reduction and crossover. Temperature works almost as an inverse age metric, with a higher value showing a newer solution and a lower value showing an older solution, albeit not quite this simple, as temperature is inherited from parents as part of crossover.

In every generation the temperature of each individual population member is reduced by a percentage amount. Fig. 5 shows when this temperature reduction takes place in the process. New members of the population are added through the crossover process, these new members inherit their temperature from their parent with the highest temperature. This inherited temperature is then increased by a percentage, meaning new members of the population will always have a higher temperature than their parents, thus increasing their priority in the sorting processes.

4.5. Hyper parameters

In optimisation algorithms there are always parameters and the proposed HS is no different. In total there are six parameters; Mutation Chance, Mutation Quantity, Initial Temperature, Temperature Modifier, Population Limit and Generation Count. The Mutation Chance is a value between 0–1 indicating how likely it is that a child solution will be mutated after crossover, with 0 representing never mutates and 1 representing it always mutates. The Mutation Quantity indicates how many genes will be manipulated across the multiple potential chromosomes in a stack. As a recommendation temperature should use the value of 1000 per chromosome within a stack. The Temperature Modifier is a percentage value that temperature is changed by every generation, or as part of crossover. For example a value of 10% would decrease the value by 10% every generation, but increase the temperature of a new population member from its hottest parent by 10%. In a population based optimisation algorithm, we need to set an upper bound for the population to reduce back down to at the end of each generation. The system must have a limit on how many generations or iterations it must complete.

There are also parameters that do not require tuning that would otherwise be expected to be set, including Crossover Point and the two IT2FLS (sorting and crossover quantity). The crossover point is set dynamically during execution using the IT2FLS dictating how much information is taken from each parent. The two IT2FLS could be modified to be problem specific, but the ones presented in this paper are made to be generic allowing them to be used in a wide variety of problems.

4.6. Balancing exploration and exploitation

The proposed Heated Stack algorithm is inspired by a genetic algorithm and more specifically NSGA-III using the ideas of population, generation, selection, domination rules and niche distance. Taking these idea and applying a temperature metric which is used to guide the system between exploration and exploitation the Heated Stack algorithm can automatically switch between exploring the search space and exploiting it. Temperature is used in two key control mechanisms for exploration and exploration, the sorting process and the crossover process, both of which guide the search space towards new front's solutions.

An Interval Type-2 Fuzzy Logic Controller is used to apply the effects of temperature in both sorting and crossover. In sorting be this for selection (deciding which member of the population mate) or for population reduction (deciding which member of the population make it through to the next generation) the effect that temperature has on exploration and exploitation is obvious. Through the use of the IT2FLC temperature is combined with the niche distance to rank the population members allowing traditionally worse performing members of the population to be higher ranked given that they have a high temperature. This higher ranking gives these members a chance to survive longer in

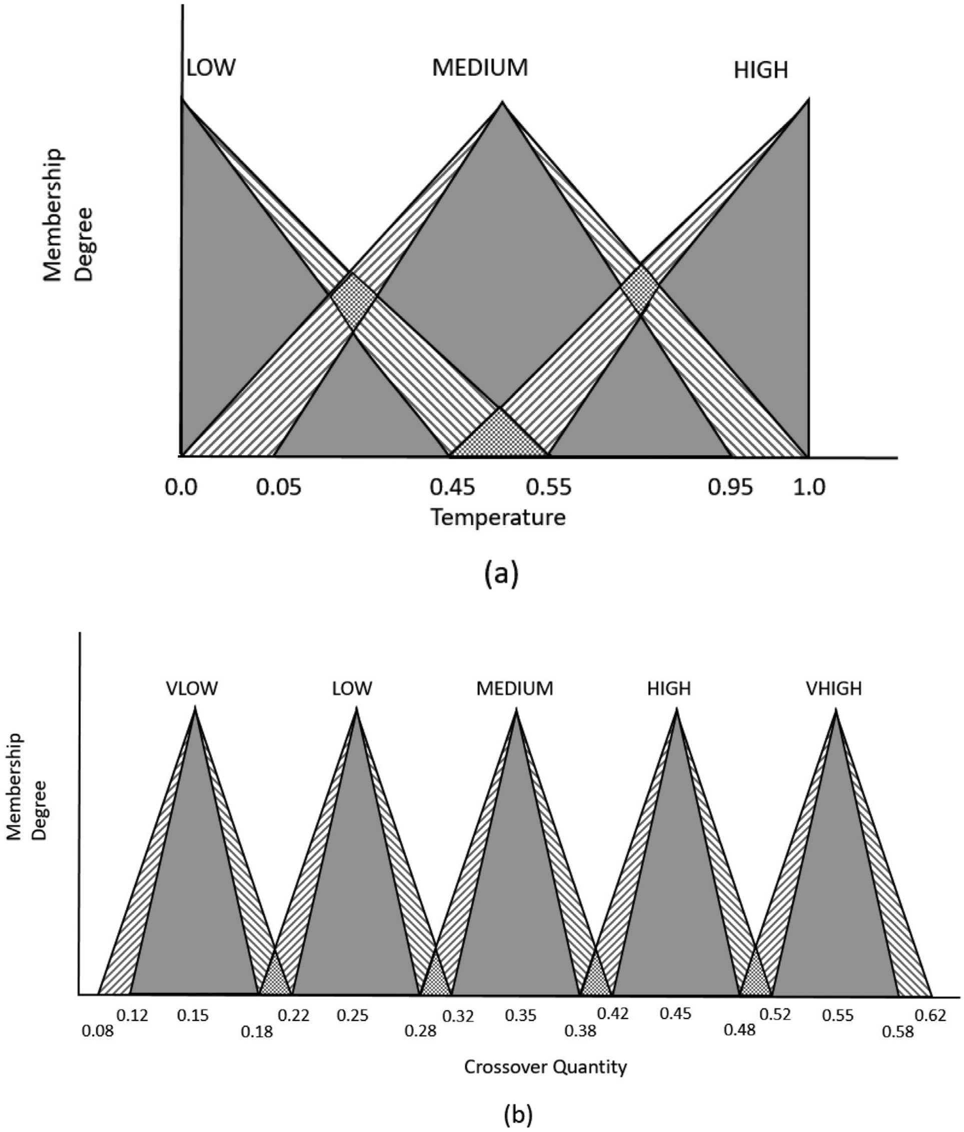

Fig. 8. (a) The input sets for temperature. (b) The output set for Crossover quantity.

the population and to mate with members of the population that are traditionally higher ranked. Once the temperature of these population members has decreased their ranking is more heavily dependent on their quality and not a mix of temperature and their quality. Therefor members of the population that have a low quality and a low temperature have less of a chance to mate with the most promising member of the population and are far more likely to be removed from the population during population reduction. As a result of the systems design in some cases more promising members of the population can sometimes be removed due a low temperature during a period of time when the average temperature within the system is very high. This is extremely unlikely to cause any significant issues as the IT2FLC primarily exists to balance the effect of temperature and quality upon the entire population.

There is second IT2FLC that uses temperature to balance exploration and exploitation in a slightly less obvious way, and works best when used in harmony with the first IT2FLC. When crossover occurs there is a comparison of temperature within the two parent solutions to determine the quantity of information to take from each parent. The theory is that a parent with a higher temperature is either newer to the population or comes from a chain of new promising solutions and thus should be prioritised to add more information to the children solutions and thus help explore a greater extent of the solution space. So the second IT2FLC uses temperature to determine how much information and thus genes should make it from each parent to the children. So when there is a larger difference in the temperature one parent is prioritised to give more information and the quantity of that information is based upon the overall temperature. Thus higher temperature population member are more likely to produce widely different offspring, whereas lower temperature population members are more likely to produce more similar offspring with minor change. In this case we can see that lower temperature crossover is more accustom to exploitation and higher temperature crossover is more alike to exploration.

Temperature is at its core used to influence the decision making processes of the proposed Heated Stack algorithm allowing it to cycle back and forth between exploration and exploitation based upon the state of the population. This population state is based upon the success of new population members and if they are better than their predecessors, as new members of the population will have a higher temperature than their parents. So with several successive generations of successful population

Table 3
List of CMOPs used in the experiments.

Name	Topic	Number of objectives	Number of constraints	Number of decision variables
RCM01	Pressure vessel design	2	2	2
RMC02	Vibrating platform design	2	5	5
RMC03	Two bar truss design	2	3	3
RCM04	Welded beam design	2	4	4
RCM05	Disc brake design	2	4	4
RCM08	Car side impact design	3	10	7
RCM17	Bulk carrier design	3	6	6
RCM18	Front rail design	2	3	3
RCM22	Haverly's pooling problem	2	6	9
RCM23	Reactor network design	2	5	6
RCM24	Heat exchanger network design	3	8	9
RCM25	Process synthesis problem	2	2	2
RCM26	Process synthesis and design problem	2	2	3

members the system will be able to pass towards a higher average temperature and start exploring. But after a few generation the average temperature will decrease and the system will exploit the solution members. Therefor temperature acts a dynamic control mechanism guiding the search space towards solutions throughout the search space.

5. Experiments & results

5.1. Experimental setup

The goal of the HS is to be an effective optimisation technique within the domain of telecommunication capacity planning, which has a low number of objectives but a high number of constraints. Due to the sensitive nature of the telecommunications domain, we are unable to go into specific detail regarding the optimisation problem. For this reason a set of problems have been taken from an open source data sources to validate the suggested systems, but to also show that HS is a capable optimisation algorithm outside of the telecoms domain and can be used in constrained multi objective problems.

Every year the conferences GECCO (The Genetic and Evolutionary Computation Conference) and CEC (Congress on Evolutionary Computation) run a wide selection of competitions including; Real-World Multi Objective Constrained Optimisation, Single Objective Bound Constrained Optimisation, Evolutionary Multi-task Optimisation, Strategy Card Game AI Competition and many more. In our experiments we have used a sub set of problems from the Real-World Multi Objective Constrained Optimisation otherwise known as Constrained Multi Objective Problems (CMOP) [46]. In this paper we have picked 13 CMOPs ranging from 2 – 3 objectives, these problems are shown in Table 3 with their associated number of objectives, constraints and decision variables.

Each of the problems shown in Table 3 is used in the comparison between the proposed HS and with NSGA-II and NSGA-III. In total there are 10 problems with 2 objectives and 3 problems with 3 objectives. NSGA-II has been selected as a comparison for this problem due to its popularity and its ability to produce high quality results in a wide variety of problems. NSGA-III has been selected for comparison, as it is the successor algorithm to NSGA-II and it is one of the two best performing algorithms in the competition.

The experimental comparison of HS vs. NSGS-II and HS vs. NAGA-III with their differing configurations are shown in Table 4. These configurations are used to show how temperature has a different effect based upon the number of iteration it has to effect the exploration and exploitation of the search. In order to keep the comparison fair NSGA-II and NSGA-III are given the same size population and the same number of generations to complete their search.

Table 4
Algorithm configuration of competition experiments.

Experiment name	Population	Generation
NSGA-II vs. HS 100	100	100
NSGA-II vs. HS 250	250	250
NSGA-II vs. HS 500	500	500
NSGA-III vs. HS 100	100	100
NSGA-III vs. HS 250	250	250
NSGA-III vs. HS 500	500	500

In the CMOP conference paper [46], they use the hyper volume indicator to compare 5 differing algorithms. In the CMOPs a baseline comparison of 7 algorithms is undertaken with NSGA-III showing it can outperform 5 of these algorithms and equal the final one. Given that NSGA-III is known to outperform a selection of high quality optimisation algorithms it was selected for comparison to HS. In this paper we have elected to use the hyper volume indicator as our metric as it is the metric used with the CMOP paper [46] and it is known to function as a high quality metric in situations with no known pareto front.

5.2. Hyper-volume indicator

As optimisation algorithms have become more successful in solving more complex problems they have become more complicated and it has become harder to evaluate the quality of an algorithm. With respect to comparing multi-objective optimisation algorithms the use of the hyper-volume indicator has become a standard measure of overall solution front quality. The hyper volume indicator also known as the Lebesgue measure [47] takes an approximated measurement of the entire searched objective space presented in the output solution front. With a high value it indicates a well explored search space with members of the final solution front not just on the far extremes of the front, but also well-spaced along it. A lower value shows a solution front that does not accommodate a large proportion of the potential search space. One way of looking at the hyper volume indicator is an estimated description of the search density provided by an optimisation algorithm's solution front.

5.3. Experimental competition problems and results

In the experiments the hyper volume indicator is used to determine which algorithm produced a better solution front. NSGA-II, NSGA-III and HS have all run each of the 13 problems 25 times in each of the three configurations in order to generate a picture of the overall quality of each algorithm at all of the presented problems. Figs. 9, 10 and 11 show which of NSGA-II or HS produced a higher hyper volume indicator on a run by run comparison and thus a better solution front, in each problem and

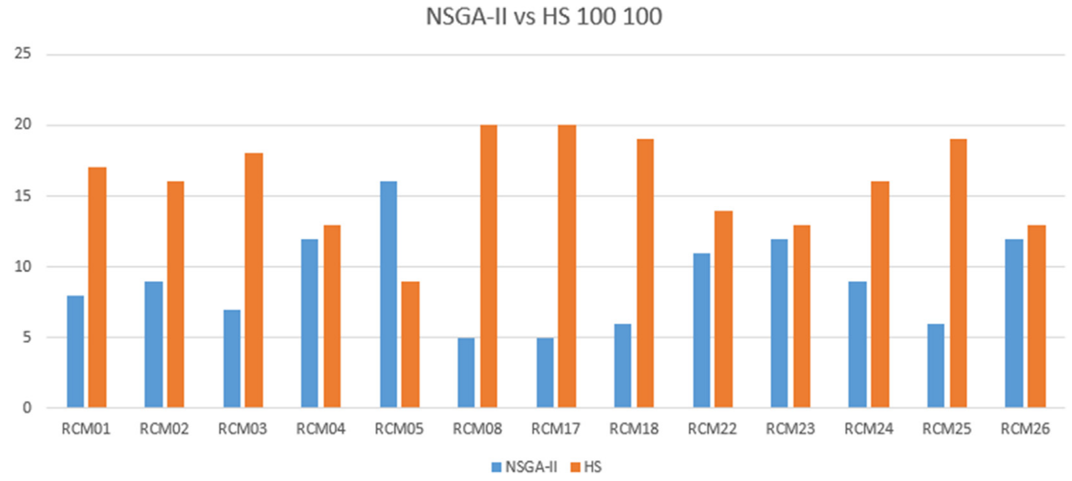

Fig. 9. A comparison of NSGA-II and HS with a configuration of 100 population members and 100 generations, indicating how many times which algorithm has the better result.

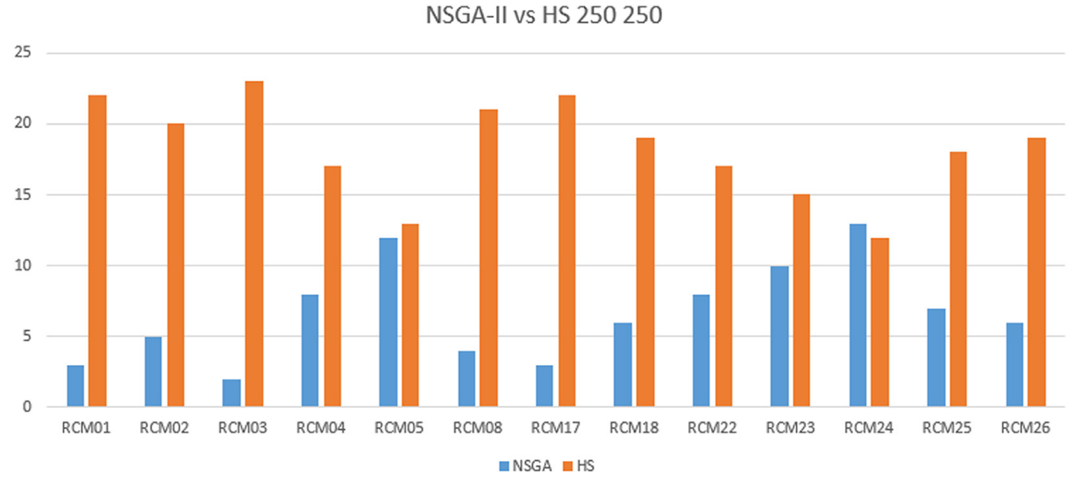

Fig. 10. A comparison of NSGA-II and HS with a configuration of 250 population members and 250 generations, indicating how many times which algorithm has the better result.

configuration. Whereas Figs. 12, 13, and 14 show the run by run comparison of NSGA-III and HS. Table 5 summarises the results shown in Figs. 11 and 14 for the CMOP experiments with a configuration of 500 generations and 500 max population, showing the median hyper volume indicator values then comparing and counting the median for HS to NSGA-II and NSGA-III respectively.

In Fig. 9 HS returns a larger hyper volume more often in 11 of the 13 problems, Fig. 10 shows an improved result for HS with 12 of the 13 problems returning a larger hyper volume more often. Fig. 11 does not offer as much as an improvement as Fig. 10, but the results still show a strong tendency for HS to produce a larger hyper volume than NSGA-II. The results indicated by Figs. 9–11 are not surprising given that HS uses a modified NSGA-III sorting methodology and NSGA-III is known to produce a better result than NSGA-II.

In Fig. 12 HS returns a larger hyper volume more often in only 4 problems. In Fig. 13 there is not much of an improvement with HS returning a larger hyper volume in only 5 problems. There is a notable improvement in Fig. 14 in how often HS has a larger hyper volume than NSGA-III with 9 of the 13 problems resulting in HS having a better result more often.

Looking at the comparisons in Figs. 9–14 alone would not draw an accurate picture of the results, so Tables 5–7 show the minimum, median, maximum and standard deviation from the CMOP experiments to give an alternate visualisation to the experimental results data. Recall that each of the three algorithms has been run on each of the 13 problems 15 times each, so there is a vast quality of results data to visualise. Specifically Table 5 show the values from the 100 population and generations experiments, Table 6 show the values from the 250 population and generation experiments and Table 7 shows the values from the 500 population and generation experiments. With the values in these Tables, we can look at the progression of the results through the differing configurations for each of the algorithms.

In the 100 generations and population experiments as seen in Table 5 when comparing HS and NSGA-II we can see that HS has a larger minimum value in 9 cases, a larger median value in 9 cases, a larger maximum value in 2 cases and a smaller standard deviation in 12 cases. What this shows us is that NSGA-II has the potential to outperform HS albeit unlikely as HS has a larger minimum and median value whilst also having a lower standard deviation. When comparing HS and NSGA-III in the 100 generations and population experiment we can see that HS has a larger minimum value in 8 cases, a larger median value in 4 cases, a larger maximum value in 5 cases and a smaller standard deviation in 8 cases. These values indicate that NSGA-III outperforms HS by a small amount due to the fact NSGA-III has a larger median and maximum value in the majority of the problems. Notably HS has a larger minimum value in more cases. The values from Table 5 match up with the conclusions of Figs. 9 and 12 indicating that

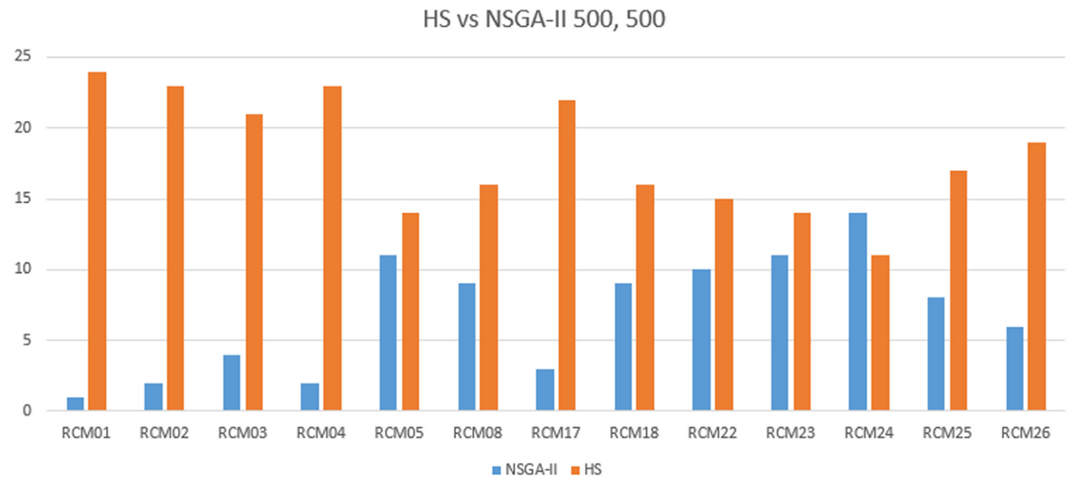

Fig. 11. A comparison of NSGA-II and HS with a configuration of 500 population members and 500 generations, indicating how many times which algorithm has the better result.

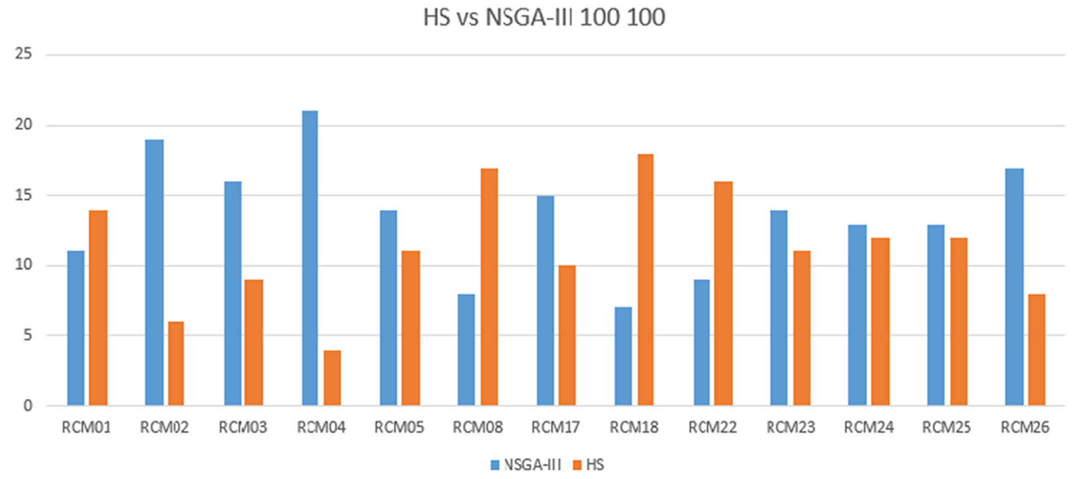

Fig. 12. A comparison of NSGA-III and HS with a configuration of 100 population members and 100 generations, indicating how many times which algorithm has the better result.

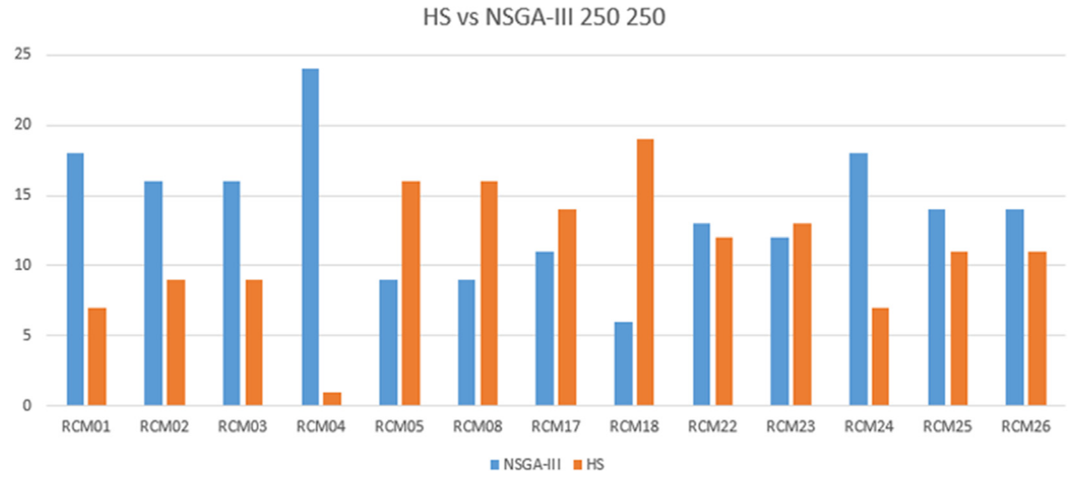

Fig. 13. A comparison of NSGA-III and HS with a configuration of 250 population members and 250 generations, indicating how many times which algorithm has the better result.

HS is outperforming NSGA-II and NSGA-III is outperforming HS for the 100 population generation experiments.

In the 250 generation and population experiment as seen in Table 6 when comparing HS and NSGA-II we can see that HS has a larger minimum value in 6 cases, a larger median value in 8 cases, a larger maximum value in 6 cases and a smaller standard deviation in 7 cases. HS is still outperforming NSGA-II but not by as much as before. This is evident due to the fact

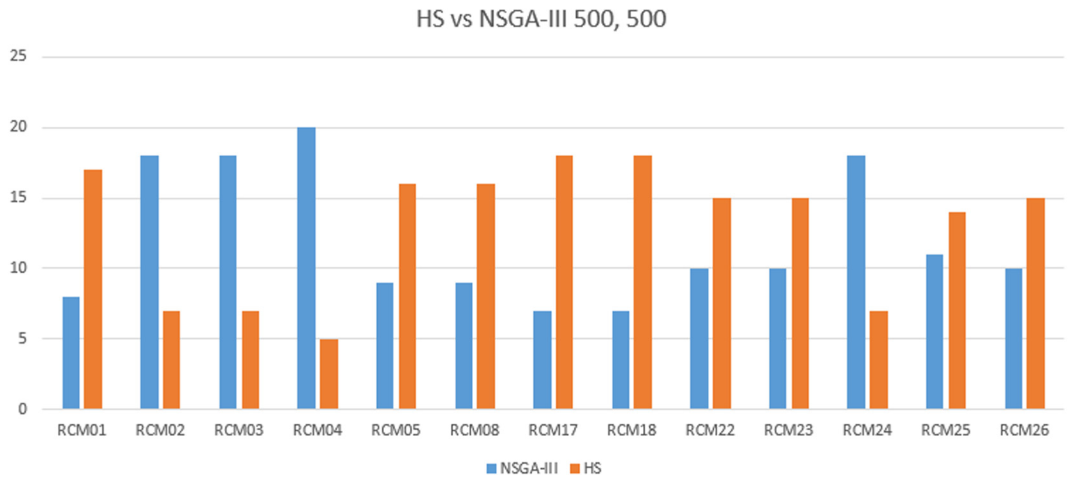

Fig. 14. A comparison of NSGA-II and HS with a configuration of 500 population members and 500 generations, indicating how many times which algorithm has the better result.

Table 5
Minimum, median, maximum and standard deviation values for HS, NSGA-II and NSGA-III in the 100 population and 100 generations experiments.

Name		Minimum	Median	Maximum	Standard deviation
RCM01	HS	4.61E+10	1.45E+12	6.03E+12	1.86E+12
	NSGA-II	1.12E+08	4.39E+09	4.52E+13	1.24E+13
	NSGA-III	5.24E+10	2.76E+11	8.04E+11	2.88E+11
RCM02	HS	2.99E+03	4.15E+04	1.89E+05	4.97E+04
	NSGA-II	6.34E+03	7.90E+03	3.96E+06	1.68E+06
	NSGA-III	3.82E+04	1.35E+06	4.13E+06	9.90E+05
RCM03	HS	1.89E+03	3.61E+04	7.81E+04	2.61E+04
	NSGA-II	3.58E+03	1.41E+04	7.89E+06	2.02E+06
	NSGA-III	1.07E+03	1.34E+06	4.24E+06	1.13E+06
RCM04	HS	2.67E+03	3.95E+05	1.06E+07	2.06E+06
	NSGA-II	2.67E+03	2.67E+03	1.29E+08	5.16E+07
	NSGA-III	1.85E+03	1.03E+05	3.06E+05	1.02E+05
RCM05	HS	3.86E+03	2.19E+04	1.88E+05	3.89E+04
	NSGA-II	1.91E+04	7.46E+04	1.67E+05	4.18E+04
	NSGA-III	1.55E+04	5.54E+04	1.01E+05	2.70E+04
RCM08	HS	8.23E+10	9.95E+11	4.08E+12	1.29E+12
	NSGA-II	3.04E+06	3.13E+10	4.42E+12	1.40E+12
	NSGA-III	2.27E+09	5.32E+11	6.85E+11	2.50E+11
RCM17	HS	5.87E+09	1.88E+11	8.17E+12	1.76E+12
	NSGA-II	4.63E+08	1.38E+11	9.76E+13	1.91E+13
	NSGA-III	4.13E+10	9.59E+11	5.26E+13	1.18E+13
RCM18	HS	7.96E+03	8.20E+03	8.90E+03	1.67E+02
	NSGA-II	7.58E+03	8.28E+03	8.97E+03	2.57E+02
	NSGA-III	7.87E+03	8.36E+03	9.01E+03	2.43E+02
RCM22	HS	2.06E+04	1.56E+05	3.02E+05	8.20E+04
	NSGA-II	1.35E+04	1.56E+05	4.68E+05	1.25E+05
	NSGA-III	1.94E+03	1.33E+05	6.31E+05	1.88E+05
RCM23	HS	1.09E+04	2.87E+04	4.75E+04	8.76E+03
	NSGA-II	1.48E+04	2.96E+04	4.24E+04	5.34E+03
	NSGA-III	1.15E+04	2.97E+04	4.35E+04	8.12E+03
RCM24	HS	6.81E+06	6.81E+06	1.96E+07	3.14E+06
	NSGA-II	8.77E+03	6.49E+06	1.12E+10	2.54E+09
	NSGA-III	3.11E+06	6.54E+07	2.01E+10	4.20E+09
RCM25	HS	8.44E+04	5.54E+05	1.10E+06	2.60E+05
	NSGA-II	4.84E+04	9.70E+05	2.96E+06	1.05E+06
	NSGA-III	4.14E+04	1.56E+06	2.43E+06	6.04E+05
RCM26	HS	2.81E+05	8.42E+05	1.75E+06	3.56E+05
	NSAG-II	2.81E+04	4.93E+05	1.91E+06	4.81E+05
	NSGA-III	7.88E+04	9.50E+05	2.93E+06	6.43E+05

the HS has a larger minimum value in fewer cases than in the 100 generation/population experiment. In this configuration by comparing the values we can see that NSGA-II appears to start performing much better, outperforming its results from the 100 generation/population experiments whereas HS does not improve much upon its 100 generation/population experiments. This lack of improvement in the HS results is how NSGA-II appears to start performing better. This lack of improvement is due to the way

Table 6
Minimum, median, maximum and standard deviation values for HS, NSGA-II and NSGA-III in the 250 population and 250 generations experiments.

Name		Minimum	Median	Maximum	Standard deviation
RCM01	HS	1.13E+09	1.43E+12	2.73E+13	5.95E+12
	NSGA-II	3.09E+08	4.25E+09	1.11E+13	2.58E+12
	NSGA-III	6.83E+10	1.91E+11	7.68E+11	2.82E+11
RCM02	HS	2.55E+03	2.87E+04	1.90E+05	4.79E+04
	NSGA-II	6.29E+03	7.90E+03	3.94E+06	7.71E+05
	NSGA-III	2.51E+03	1.60E+06	2.01E+06	6.81E+05
RCM03	HS	1.88E+03	4.26E+04	1.90E+05	4.73E+04
	NSGA-II	2.06E+03	7.90E+03	4.20E+05	1.02E+05
	NSGA-III	2.67E+03	5.68E+05	2.03E+06	7.69E+05
RCM04	HS	2.67E+03	1.45E+04	7.91E+06	1.67E+06
	NSGA-II	2.67E+03	2.67E+03	1.29E+08	5.13E+07
	NSGA-III	1.74E+03	2.77E+05	2.84E+05	9.96E+04
RCM05	HS	7.12E+03	4.37E+04	9.16E+05	2.89E+05
	NSGA-II	1.36E+04	5.94E+04	9.58E+04	2.58E+04
	NSGA-III	1.12E+04	3.74E+04	9.50E+04	2.45E+04
RCM08	HS	1.16E+09	1.43E+12	4.41E+12	1.26E+12
	NSGA-II	4.04E+08	1.56E+11	4.41E+12	8.57E+11
	NSGA-III	3.35E+09	5.21E+11	1.22E+12	3.15E+11
RCM17	HS	7.77E+09	5.07E+11	2.50E+13	5.03E+12
	NSGA-II	8.68E+09	8.43E+10	3.92E+12	7.90E+11
	NSGA-III	6.04E+08	2.09E+10	4.54E+11	1.08E+11
RCM18	HS	8.08E+03	8.26E+03	8.43E+03	1.09E+02
	NSGA-II	8.14E+03	8.37E+03	9.04E+03	1.96E+02
	NSGA-III	7.55E+03	7.91E+03	8.94E+03	2.65E+02
RCM22	HS	3.84E+03	7.38E+04	2.49E+05	7.03E+04
	NSGA-II	5.34E+03	1.13E+05	5.64E+05	1.22E+05
	NSGA-III	2.72E+04	1.84E+05	5.77E+05	1.43E+05
RCM23	HS	1.09E+04	2.48E+04	4.36E+04	8.50E+03
	NSGA-II	2.03E+04	2.91E+04	3.53E+04	2.83E+03
	NSGA-III	1.68E+04	2.74E+04	4.23E+04	6.00E+03
RCM24	HS	6.81E+06	6.81E+06	3.81E+07	8.31E+06
	NSGA-II	1.95E+06	3.46E+06	2.65E+11	5.25E+10
	NSGA-III	6.03E+05	2.72E+08	1.63E+11	3.45E+10
RCM25	HS	2.16E+05	6.38E+05	1.42E+06	3.61E+05
	NSGA-II	4.95E+04	9.70E+05	2.95E+06	8.11E+05
	NSGA-III	2.93E+04	1.24E+06	2.22E+06	6.00E+05
RCM26	HS	2.01E+05	1.12E+06	1.69E+06	3.85E+05
	NSGA-II	1.04E+05	7.11E+05	1.39E+06	3.72E+05
	NSGA-III	1.01E+05	1.12E+06	2.12E+06	5.31E+05

that temperature interacts with exploration and exploitation, but more on this later. When comparing HS and NSGA-III we can see that HS has a larger minimum value in 7 cases, a larger median value in 6 cases, a larger maximum value in 6 cases and a smaller standard deviation in 7 cases. These values show us that NSGA-III and HS are comparable at this configuration with neither of the algorithms showing a clear advantage over the other. The values show a similar result to that of Fig. 10 although Table 6 shows in more detail that the gap between NSGA-II and HS is closer than before, with NSGA-II having a vast improvement upon its 100 generation/population results, whereas the improvement logged by HS is less prominent and even regressing in some cases. We can also see the values from Table 6 show similar results to Fig. 13 with NSGA-III having the slightest advantage over HS although the values are so close there is no clear cut winner. Interestingly by comparing the NSGA-III values for the 100 generation/population experiment to the corresponding values in the 250 generation/population experiment we can see that there is little to no improvement and in some cases the 100 generation population experiment appears to perform better.

In the 500 generation and population experiment as seen in Table 7 when comparing HS and NSGA-II we can see that HS has a larger minimum value in 10 cases, a larger median value in 11 cases, a larger maximum value in 8 cases and a smaller standard deviation in 12 cases. These values show that HS is more likely to get a higher value in the 500 generation/population experiments and present a better solution front than NSGA-II more often. This is due to HS having larger median value more often along with its smaller standard deviation and its higher minimum and maximum values. There are still some cases where NSGA-II performs exceptionally well and beats HS and NSGA-III such as the maximum value for problem RCM23. When comparing HS and NSGA-III we can see that HS gets a larger minimum value in 8 cases, a larger median value in 9 cases, a larger maximum and a smaller standard deviation in 12 cases. By looking at these values together it can be seen that HS is more likely to get a higher value more often with its higher minimum and median values and its lower standard deviation. The values from Table 7 show a corresponding result to that presented in Fig. 11 with HS outperforming NSGA-II the majority of the time additionally we can see the same results as that displayed in Fig. 14 with HS being more likely to produce a better result than NSGA-III but NSGA-II can still outperform HS on occasion.

The idea of the results presented in Figs. 9–14 and Tables 5–7 is to indicate which of the three algorithms can produce the best solution front across all of the experiments the most consistently. In order to produce these Tables and figures a set of experiments was run for each of the algorithms. There are 13 problems selected from the CMOP competitions run by GECCO and CEC [46]. Each of these problems is run a total of 25 times. So

Table 7
Minimum, median, maximum and standard deviation values for HS, NSGA-II and NSGA-III in the 500 population and 500 generations experiment.

Name		Minimum	Median	Maximum	Standard deviation
RCM01	HS	2.83E+09	2.15E+11	7.38E+12	2.33E+12
	NSGA-II	4.18E+08	4.25E+08	1.84E+10	1.07E+13
	NSGA-III	4.84E+09	1.32E+11	8.10E+11	1.06E+13
RCM02	HS	4.43E+03	2.00E+05	1.03E+05	2.71E+04
	NSGA-II	3.43E+03	7.90E+03	1.12E+05	1.39E+06
	NSGA-III	2.76E+03	1.41E+06	1.97E+06	1.37E+06
RCM03	HS	5.91E+02	7.06E+05	2.08E+05	5.49E+04
	NSGA-II	5.43E+02	7.90E+03	1.68E+06	1.83E+06
	NSGA-III	5.83E+03	4.32E+04	2.58E+06	1.80E+06
RCM04	HS	2.99E+03	6.84E+04	7.93E+07	1.55E+07
	NSGA-II	9.26E+02	2.60E+03	5.90E+04	3.91E+07
	NSGA-III	2.18E+03	2.53E+05	2.99E+05	3.87E+07
RCM05	HS	5.14E+03	4.98E+04	1.80E+06	4.48E+05
	NSGA-II	1.18E+04	5.81E+04	1.57E+05	4.01E+05
	NSGA-III	1.18E+04	3.70E+04	1.46E+05	3.94E+05
RCM08	HS	1.23E+10	4.35E+11	5.68E+12	1.70E+12
	NSGA-II	5.87E+08	5.88E+10	4.42E+12	1.81E+12
	NSGA-III	1.10E+06	5.44E+11	9.92E+11	1.82E+12
RCM17	HS	8.61E+09	1.82E+13	1.06E+13	2.76E+12
	NSGA-II	6.06E+06	5.16E+11	1.29E+11	2.26E+13
	NSGA-III	8.22E+10	3.13E+12	7.29E+14	2.23E+13
RCM18	HS	8.08E+03	8.46E+03	9.42E+03	2.44E+02
	NSGA-II	8.08E+03	8.39E+03	9.07E+03	1.78E+03
	NSGA-III	7.59E+03	8.35E+03	1.01E+04	2.16E+03
RCM22	HS	1.53E+05	2.23E+05	2.66E+05	2.61E+04
	NSGA-II	4.40E+03	8.47E+04	4.44E+05	1.94E+05
	NSGA-III	2.30E+04	1.70E+05	4.09E+05	1.96E+05
RCM23	HS	1.22E+04	2.98E+04	5.31E+04	1.21E+04
	NSGA-II	1.95E+04	8.41E+04	4.20E+04	1.30E+04
	NSGA-III	1.11E+04	2.97E+04	4.07E+04	1.24E+04
RCM24	HS	7.63E+06	3.57E+07	7.63E+07	1.86E+07
	NSGA-II	5.96E+05	2.02E+04	1.87E+11	6.32E+10
	NSGA-III	3.17E+06	5.66E+08	9.51E+09	7.06E+10
RCM25	HS	2.37E+05	1.50E+06	1.54E+06	3.35E+05
	NSGA-II	4.95E+04	9.70E+05	4.91E+06	9.18E+05
	NSGA-III	5.66E+05	1.13E+06	4.44E+06	1.19E+06
RCM26	HS	1.19E+05	1.41E+06	6.93E+06	1.29E+06
	NSGA-II	1.60E+05	8.96E+05	3.68E+06	1.46E+06
	NSGA-III	5.31E+04	1.28E+06	2.46E+06	1.51E+06

in total 975 experiments were run to produce the results for the CMOP experiments. To summarise the idea of a consistent result we shall just look at the median values of the 500 generations and 500 population experiment. In 11 out of the 13 cases or 84.6% HS results in a better median hyper-volume indicator value than NSGA-II. Whereas comparing HS and NSGA-III the median results show that HS gives a better value in 9 out of the 13 experiments or 69.2% of the time. Interestingly in 3 cases NSGA-II has a better result than NSGA-III when we look at the median values.

5.4. Capacity planning results

The HS algorithm was created to solve a specific capacity planning problem within the telecoms exchange. The problem is comprised of two objectives, with four constraints using 1200 decision variables in a multi-chromosomal solution representation. Figs. 15 and 16 show a comparison of which algorithm at which configuration gave the better hyper volume indicator more often. Fig. 15 shows a comparison of NSGA-II and HS with the HS showing that it clearly provides a solution front with a high hyper volume more often. The number of experiments that HS outperformed NSGA-II increases as the maximum population and number of generation increases. This increase is to ensure the intended effect that temperature has on the search space over time, as temperature increases and decreases it guides HS between being explorative to exploitative.

Fig. 16 shows the comparison between NSGA-III and HS at different parameter configurations, in the first case (100, 100) NSGA-III outperforms HS drastically. In the second case (250, 250) HS begins to outperform NSGA-III giving a larger hyper volume more often. Finally HS further outperforms NSGA-III in the last configuration (500, 500) by a larger margin. Once again this margin is due to HS having more iterations to allow temperature to guide the search.

Looking at Table 8, we can see it reinforces the results shown by Fig. 15 with HS performing better than NSGA-II in all configurations, with a larger minimum, median and maximum values. Table 8. also reinforces the results shown in Fig. 16 with HS performing worse the NSGA-III in the first configuration with a lower minimum and median values, although HS does have a higher maximum value. In the 250 generation/population & 500 generation/population configurations HS still has a worse minimum value than NSGA-III but its median and maximum values are considerably higher, showing that HS has improved upon NSGA-III as it has more iterations.

The CMOP and Telecoms experiments show a similar outcome, that as the number of iterations increase the proposed HS performs better. This increase is an expected behaviour as the HS uses temperature as a search heuristic allowing for worse

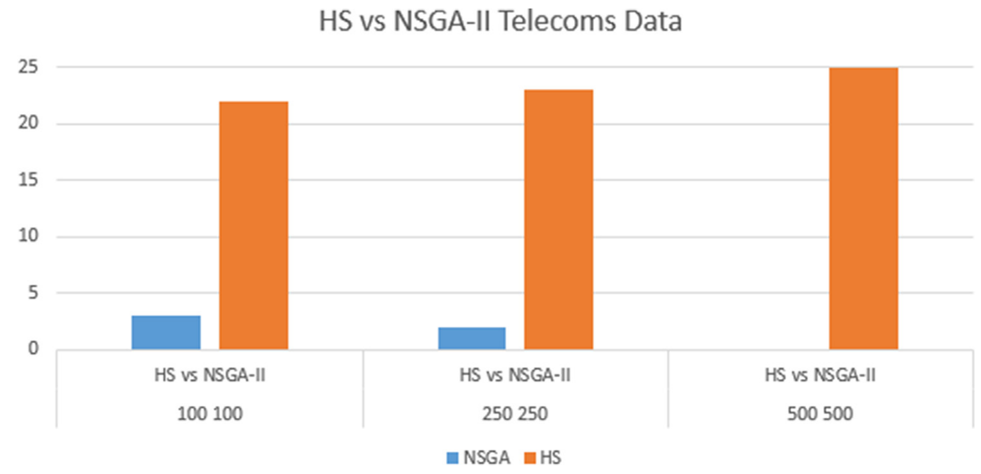

Fig. 15. Comparison of NSGA-II and HS at the telecoms problem, with differing population and generation parameters.

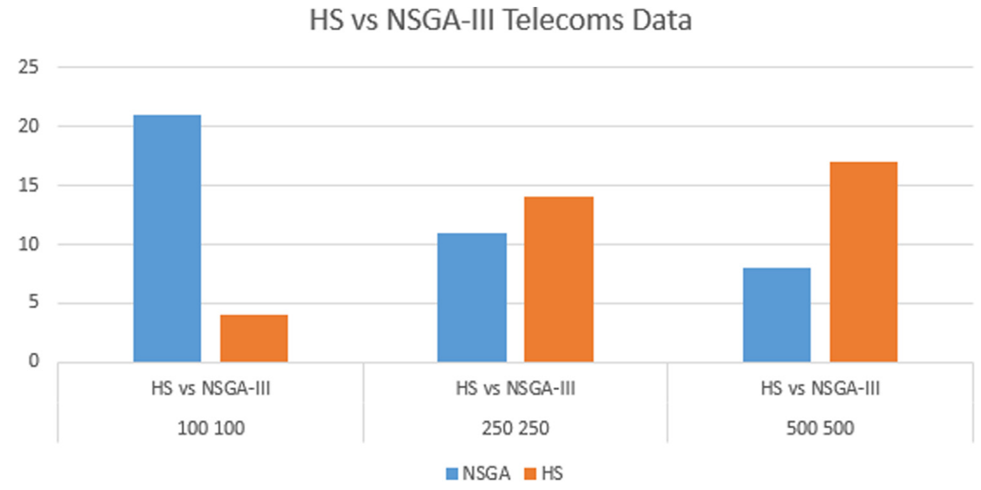

Fig. 16. Comparison of NSGA-III and HS at the telecoms problem, with differing population and generation parameters.

Table 8
Comparison of minimum, median maximum and standard deviation values of different configurations for the telecoms problem.

Configuration	Algorithm	Minimum	Median	Maximum	Standard deviation
100, 100	HS	3.26E+12	3.43E+12	7.03E+12	1.27E+12
	NSGA-II	3.16E+12	3.31E+12	3.37E+12	5.06E+10
	NSGA-III	3.74E+12	3.89E+12	3.97E+12	5.13E+10
250, 250	HS	3.08E+12	6.25E+12	6.97E+12	1.59E+12
	NSGA-II	3.02E+12	3.11E+12	3.20E+12	5.02E+10
	NSGA-III	3.70E+12	3.84E+12	3.91E+12	5.11E+10
500, 500	HS	3.10E+12	6.29E+12	7.08E+12	1.68E+12
	NSGA-II	2.86E+12	2.99E+12	3.06E+12	5.67E+10
	NSGA-III	3.72E+12	3.78E+12	3.85E+12	4.03E+10

performing members of the population to contribute to the search before being removed from the population. Temperature requires iterations to go through the difference stages of the IT2FLS allowing it to use the full range of the rule base. The interaction between the addition of new members of the population and their temperature allows the HS to propagate temperature increase throughout the population whilst also reducing the temperature of population members over a number of generations. This interaction of temperature allows the system to transfer back and forth between explorative to exploitative phases. With the system being explorative when the members of the population have a high temperature and exploitative when they have a low temperature.

6. Conclusions & future work

The capacity planning problem within the telecoms industry is not a simple problem to solve, it consists of optimising a multi-layered structure with interconnected objectives and constraints. The multilayer structure is made up of bandwidth hardware including: switches, cards and ports. The solutions presented by the Heated Stack (HS) for the capacity planning problem has a lasting real-world effect on national infrastructure, therefore it is important to get a high quality solution which meets the objectives and constraints. There have been previous attempts to solve this sort of problem across multiple domains which have been successful within their domains but are not capable of solving the telecoms version of the capacity planning problem.

HS is capable of solving this problem to a high standard when compared to two of the best general optimisation algorithms NSGA-II and NSGA-III. When HS is used for the telecoms capacity planning problem it outperforms NSGA-II 100% of the time over the course of 25 experiments, and outperforms NSGA-III 68% of the time over the course of 25 experiments.

To proof HS capability as a powerful optimisation systems, we also ran a sub-set of the constrained multi-objective problems from the competitions presented by the IEEE Congress on Evolutionary Computation in order to give more transparency to the experiments, as well as showing the general optimisation capabilities of HS. These problems involved between two and three objectives, and between two and seven constraints. Looking at the median hyper volume indicator values across 13 problems each run for 25 experiments HS provides a better resulting solution front in 84.6% of cases when compared to NSGA-II whereas when compared to NSGA-III using same metric, HS provides a better resulting solution front in 69.2% of cases.

These results are due to the core idea of HS — temperature, where the system can change between an explorative state and exploitive state based upon the rate at which new member of the population are created. In our previous work we indicated that there is an improvement by using a Type-2 Fuzzy Logic System (FLS) over a Type-1 FLS or a Crips numbering system within the HS to interact with the temperature metric [45]. In this paper we have expanded upon this previous work allowing for a wider exploration of open source problems, whilst also providing a new more complex capacity planning problem from within the Telecommunication Industry. In these problems we have seen an improvement over NSGA-II and NSGA-III. In the future we are intending to create explainable optimisation solutions using backwards induction propagated through the use of Monte-Carlo Tree Search from the optimisation results provided from HS.

CRediT authorship contribution statement

Lewis Veryard: Primary author, Did most of conceptualising, programming and writing. **Hani Hagras:** Ph.D. supervisor to Lewis Veryard, Given support and guidance throughout the process of conceptualising and writing. **Anthony Conway:** British Telecom supervisor, Provided data and practical programming advise to produce a working prototype. **Gilbert Owusu:** British Telecom Management, Provided guidance and support.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Lewis Veryard's Ph.D. is fully funded by British Telecom (BT) and by extension so is this piece of research. By the time of submission BT are using the research in its industrial context, with plans to expand upon it. This work has not be submitted anywhere else, or is it under review by any other publisher.

Data availability

The data that has been used is confidential.

References

[1] Cisco, Cisco: 2020 CISO benchmark report, 2020.

[2] H. Fujiwara, K.M. Kobayashi, Bin packing with cardinality constraints, in: M.-Y. Kao (Ed.), Encyclopedia of Algorithms, Springer US, Boston, MA, 2008, pp. 1–4.

[3] E. Tomita, Knapsack, in: M.-Y. Kao (Ed.), Encyclopedia of Algorithms, Springer US, Boston, MA, 2014, pp. 1–4.

[4] N. Yusupova, D. Rizvanov, E. Chernyshev, Evaluation of the effectiveness of the multi-agent approach for capacity planning, in: Proc. - ICOECS 2020 2020 Int. Conf. Electrotech. Complexes Syst, 2020.

[5] Y.K. Lu, W.Y. Chen, Implementation of required capacity planning for virtual machine development, in: Proc. 2012 8th Int. Conf. Intell. Inf. Hiding Multimed. Signal Process. IIH-MSP 2012, 2012, pp. 331–334.

[6] S.G. Hamed, M.S. Pishvaee, A system dynamics approach for capacity planning and price adjustment in a closed-loop supply chain, in: EMS 2009 - UKSim 3rd Eur. Model. Symp. Comput. Model. Simul. (2), 2009, pp. 435–439.

[7] K. Kurihara, H. Maruyama, K. Masuda, Hierarchical planning method for product supply based on multi objective genetic algorithm, in: Proc. 15th IEEE Int. Conf. Emerg. Technol. Fact. Autom. ETFA 2010, 2010, pp. 1–8.

[8] O. Gerstel others, Multi-layer capacity planning for IP-optical networks, in: IEEE Communications Magazine, Vol. January, IEEE, 2014, pp. 44–51.

[9] P. Papanikolaou, K. Christodoulopoulos, E. Varvarigos, Incremental planning of multi-layer elastic optical networks, in: 2017 21st Int. Conf. Opt. Netw. Des. Model. ONDM 2017 - Conf. Proc, 2017.

[10] C.A.A. Herrera, D. Özdemir, M. Cabrera-Ríos, Capacity planning in a telecommunications network: A case study, Int. J. Ind. Eng. Theory Appl. Pract. 16 (2) (2009) 82–90.

[11] Y. D'Halluin, P. a Forsyth, K.R. Vetzal, Managing Capacity for Telecommunications Networks Under Uncertainty, 10 (4) (2002) 579–588.

[12] C.M.A. García, A.M.Á. Socarrás, Approaching a capacity expansion problem in telecommunications, in: Proc. 16th IEEE Int. Conf. Electron. Commun. Comput. CONIELECOMP 2006, 2006, No. Conielecomp, 2006, p. 1.

[13] B. Chopard, M. Tomassini, Particle Swarm Optimization, in: Natural Computing Series, 2018.

[14] O.K. Erol, I. Eksin, A new optimization method: Big Bang-Big Crunch, Adv. Eng. Softw. 37 (2) (2006) 106–111.

[15] M. Dorigo, M. Birattar, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.

[16] J.H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–72.

[17] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science (80-.). 220 (4598) (1983) 671–680.

[18] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. (21) (1953) 1087–1092.

[19] K. Sharp, F. Matschinsky, Translation of ludwig Boltzmann's paper 'on the relationship between the second fundamental theorem of the mechanical theory of heat and probability calculations regarding the conditions for thermal equilibrium' Sitzungberichte der kaiserlichen akademie d, Entropy 17 (4) (2015) 1971–2009.

[20] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. 22 (4) (2018) 609–622.

[21] Z.Z. Liu, Y. Wang, P.Q. Huang, AnD: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation, Inf. Sci. (Ny). 509 (2020) 400–419.

[22] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, IEEE Trans. Evol. Comput. 25 (1) (2021) 102–116.

[23] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[24] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.

[25] V. Chankong, Y.Y. Haimes, Multiobjective Decision Making Theory and Methodology, North-Holland, New York, NY, USA, 1983.

[26] M. Fleischer, The measure of Pareto optima: Applications to multi-, 2002.

[27] N. Srinivas, K. Deb, Muiltiobjective optimization using nondominated sorting in genetic algorithms, Evol. Comput. (1994).

[28] I. Das, J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.

[29] H. Jain, K. Deb, An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach, IEEE Trans. Evol. Comput. 18 (4) (2014) 602–622.

[30] N. Cesa-Bianchi, Multi-armed bandit problem, in: M.-Y. Kao (Ed.), Encyclopedia of Algorithms, Springer US, Boston, MA, 2014, pp. 1–5.

[31] M.N. Katehakis, A.F. Veinott, Multi-Armed Bandit problem: Decomposition and computation, Math. Oper. Res. 12 (2) (1987) 262–268.

[32] L.A. Zadeh, Fuzzy sets, Inf. Control 8 (1965) 338–353.

[33] J.M. Mendel, Uncertain Rule-Based Fuzzy Systems, Springer US, 2017.

[34] H.A. Hagras, A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots, IEEE Trans. Fuzzy Syst. 12 (4) (2004) 524–539.

[35] J.M. Mendel, R.I.B. John, Type-2 fuzzy sets made simple, IEEE Trans. Fuzzy Syst. 10 (2) (2002) 117–127.

[36] J.M. Mendel, Advances in type-2 fuzzy sets and systems, Inf. Sci. (Ny) (2007).

[37] A. Sakalli, T. Kumbasar, E. Yesil, H. Hagras, Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system, in: IEEE Int. Conf. Fuzzy Syst, 2014, pp. 1859–1866.

[38] H. Leon-Garza, H. Hagras, A. Pena-Rios, A. Conway, G. Owusu, A big bang-big crunch type-2 fuzzy logic system for explainable semantic segmentation of trees in satellite images using HSV color space, in: IEEE Int. Conf. Fuzzy Syst. 2020-July, 2020.

[39] L. Veryard, H. Hagras, A. Conway, G. Owusu, A type-2 fuzzy genetic approach to uncertain & dynamic resilient routing within telecommunications networks, in: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2020.

[40] A. Bilgin, H. Hagras, A. Malibari, M.J. Alhaddad, D. Alghazzawi, Towards a general type-2 fuzzy logic approach for Computing with Words using linear adjectives, in: IEEE International Conference on Fuzzy Systems, 2012.

[41] L. Beasley, H. Hagras, A. Conway, G. Owusu, A type-2 fuzzy based multi-objective optimisation for strategic network planning in the telecommunication domain, in: IEEE Int. Conf. Fuzzy Syst. 2021-July, 2021, pp. 1–7.

[42] Y. H, Z.G. Lu J, Topology learning-based fuzzy random neural network for streaming data regression, IEEE Trans. Fuzzy Syst. (2020).

[43] A. Takahashi, S. Takahashi, A new interval type-2 fuzzy logic system under dynamic environment: Application to financial investment, Eng. Appl. Artif. Intell. (2021).

[44] L. Lv, J. Wang, J. Long, Interval type-2 fuzzy logic anti-lock braking control for electric vehicles under complex road conditions, Sustain. 13 (20) (2021).

[45] L. Veryard, H. Hagras, A. Conway, G. Owusu, A type-2 fuzzy multi-objective multi-chromosomal optimisation for capacity planning within telecommunication networks, in: IEEE Int. Conf. Fuzzy Syst. 2021-July, 2021.

[46] A. Kumar others, A benchmark-suite of real-world constrained multi-objective optimization problems and some baseline results, Swarm Evol. Comput. 67 (2021).

[47] H. Lebesgue, Intégrale, Longueur, Aire, Université de Paris, 1902.