processes

MDPI

Article

A Tandem Running Strategy-Based Heat Transfer Search Algorithm and Its Application to Chemical Constrained Process Optimization

Ebrahim Alnahari and Hongbo Shi *

Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China; barhooma97@hotmail.com

* Correspondence: hbshi@ecust.edu.cn; Tel.: +86-21-6425-2189

Abstract: Constrained optimization problems (COPs) are widely encountered in chemical engineering processes, and are normally defined by complex objective functions with a large number of constraints. Classical optimization methods often fail to solve such problems. In this paper, to solve COPs efficiently, a two-phase search method based on a heat transfer search (HTS) algorithm and a tandem running (TR) strategy is proposed. The main framework of the MHTS–TR method aims to alternate between a feasible search phase that only examines feasible solutions, using the HTS algorithm, and an infeasible search phase where the treatment of infeasible solutions is relaxed in a controlled manner, using the TR strategy. These two phases play different roles in the search process; the former ensures an intensified optimum in a relevant feasible region, whereas the latter is used to introduce more diversity into the former. Thus, the ensemble of these two complementary phases can provide an effective method to solve a wide variety of COPs. The proposed variant was investigated over 24 well-known constrained benchmark functions, and then compared with various well-established metaheuristic approaches. Furthermore, it was applied for solving a chemical COP. The promising results demonstrate that the MHTS–TR approach is applicable for solving real-world COPs.

Keywords: chemical processes; constrained optimization; engineering design problems; heat transfer search algorithm; tandem running strategy; global optimization; constraint-handling techniques

check for updates

Citation: Alnahari, E.; Shi, H. A Tandem Running Strategy-Based Heat Transfer Search Algorithm and Its Application to Chemical Constrained Process Optimization. *Processes* **2021**, *9*, 1961. https://doi.org/10.3390/pr9111961

Academic Editor: Xu Ji

Received: 7 September 2021
Accepted: 28 October 2021
Published: 2 November 2021

1. Introduction

Nowadays, many real-world chemical engineering processes are defined by complex objective functions with a large number of constraints [1]. The optimization problems that contain several constraints are described as constrained optimization problems (COPs) [2]. These problems are generally characterized by their different types, such as linear, nonlinear, polynomial, quadratic, cubic, etc. Due to the complexity of highly constrained chemical processes, new generation optimization methods need to be found, as classical methods often fail to solve COPs efficiently. Thus, various metaheuristic algorithms (MHAs) have been developed, modified, and applied significantly to optimize a wide variety of COPs [2–4].

Various approaches have been applied as constraint-handling techniques to deal with COPs during the search course, as reported in the surveys [5,6]. One of the most popular approaches are penalty-based methods [7–9], which can be simply classified into static and self-adaptive methods. The static methods deal with the infeasible solutions, by transforming a COP into an unconstrained problem. However, defining the penalty parameter values is not straightforward. Self-adaptive penalty methods modify the penalty term value throughout the search course, such as the adaptive penalty method (APM) [10], an effective penalty-based method that automatically calibrates the infeasible surface throughout evolution. However, it may lose feasible solutions during the search course. A set of three feasibility rules called Deb's rules [11] is a popular selecting candidate solutions technique, using pre-established priority rules which separately rank the solutions depending on their constraint violation values and objective function values, respectively. These rules state that: (i) between two feasible solutions, the solution that has a superior value of the objective function is preferred; (ii) between feasible and infeasible solutions, the feasible one is preferred; (iii) between two infeasible solutions, the one with smaller constraint violation values is preferred. These rules can be incorporated with several MHAs, since they do not often require any additional parameters [12]. A variant of the feasibility rules for diversity maintenance was proposed by Mezura and Coello [13]. It is one of the simplest mechanisms that allows a set of infeasible solutions to remain in the search. In this variant, these solutions are adjacent to the feasible domain; moreover, they have a great objective function value and the lowest sum of constraint violation. These solutions are selected from the offspring or the parent's population with a 50% probability.

When solving complex COPs, simply finding a feasible solution is not a straightforward task. Several types of research on highly constrained optimization problems [14–16] have reported that the consideration of infeasible solutions during the search course, instead of limiting the search process to feasible regions, may help to better explore the search space. As it can be noted from the above-mentioned works of literature, the main goal of their work is the focus on the addition of evolutionary mechanisms within the feasible and infeasible regions during the search process. The idea of preserving the infeasible solutions close to the feasible region allows for finding an optimum on the boundary of the feasible region of the search space [17]. The optimal solution of COPs is often located in the interior or near the boundary of the feasible region. Hence, the individuals that are far away from the feasible region are practically unhelpful to the optimization of the population, whereas the individuals that are close to the feasible region may contain favorable information that can help the population in searching for an optimum on or near the boundary of the feasible region. Therefore, based on the above observation, it is important to consider how to properly preserve the infeasible individuals close to the feasible region during the search course, for the sake of guiding the search of the feasible region toward the global optima of COPs.

The heat transfer search (HTS) algorithm [18] is a novel population-based method inspired by the natural laws of thermodynamics and heat transfer. Its basic framework mainly includes three main phases such as conduction, convection, and radiation. This algorithm belongs to the family of new generation MHAs, and is considered one of the most competent optimization approaches in comparison with other MHAs, as shown in the literature [18]. Although it is a relatively new method, a number of its variants have been developed, modified, and applied significantly to solve a wide variety of optimization problems [19–26]. However, the HTS-based variants have not been applied for handling chemical COPs. Therefore, we aim to extend the actual applications of this MHA to handle such problems.

The major contributions of our paper are described below:

(1) A novel approach with two search phases called MHTS–TR is proposed through integrating the multiple HTS algorithm and the TR strategy. The ensemble of these two complementary phases can provide an effective algorithm for solving various COPs;
(2) The effectiveness of the new variant is tested through a set of 24 constrained benchmark problems, and the simulation results are compared with those of other competitors;
(3) The MHTS–TR approach is used to handle a real-world chemical COPs. Additionally, the simulation results obtained on this problem are compared with those of different approaches existing in the literature. To the best of our knowledge, this paper presents the first attempt for applying an HTS-based method to handle a chemical COP.

The rest of our work is organized as follows: the main formula of COPs is defined in Section 2; the main theoretical principles of the TR strategy and HTS method are described in Section 3; the new variant is explained in detail in Section 4; in Section 5, certain experimental investigations and comparisons are conducted, and the proposed MHTS–

TR approach is used for solving a real-world COP in Section 6. In Section 7, our final conclusions are summarized.

2. Problem Statement

In general, the mathematical model of a COP can be described as follows, where the main goal is to optimize the objective function, represented as $f(x)$:

$$\begin{array}{ll} \text{minimize} & f(x),\ x = [x_1, \ldots, x_i, \ldots, x_n] \in R^n \\ \text{subject to} & g_j(x) \le 0,\ (j = 1, \ldots, l) \\ & h_j(x) = 0,\ (j = l+1, \ldots, p) \\ & x_i^{(Low)} \le x_i \le x_i^{(Up)},\ (i = 1, 2, \ldots, n) \end{array} \tag{1}$$

where $f(x)$ represents the fitness function; $x \in \Omega \subseteq S$ indicates the n-dimensional solution vector, x_i denotes the ith dimensional component of x; $S \in R^n$ indicates the solution space determined by the upper and lower bounds ($x^{max} = \left[{x_1}^{max}, \ldots, {x_i}^{max}, \ldots, {x_n}^{max}\right]$ and $x^{min} = \left[{x_1}^{min}, \ldots, {x_i}^{min}, \ldots, {x_n}^{min}\right]$) of the solution vector x; Ω represents the feasible region of dimension n; $g_j(x) \le 0$ indicates the inequality constraint; $h_j(x) = 0$ denotes the equality constraint, and l and p are defined as the number of inequality and equality constraints, respectively. Due to the constraints shown in Equation (1), two disjoint subsets (feasible and infeasible) constitute the search domain. The feasible domain is defined by the regions where all p constraint functions of equalities and inequalities are satisfied. Thus, the solutions x belonging to the feasible region and infeasible region are classified as feasible and infeasible candidate solutions, respectively.

In general, the constraint-handling techniques can be classified either as indirect, when both feasible and infeasible candidate solutions are considered along the search, or as direct, when only the feasible candidate solutions are employed. The penalty method is the most common indirect approach applied with MHAs to penalize the infeasible solutions. In this method, when x is an infeasible solution, its objective function is penalized by adding a penalty term, which depends on the constraint violation. When solving COPs, in addition to calculating the fitness value of the population according to the function $f(x)$, it is also necessary to evaluate the constraint violation. Generally, the violation degree of a member x to the jth constraint can be expressed as follows:

$$G_j(x) = \begin{cases} \max\{0,\ g_j(x)\} \ \text{if } 1 \le j \le l \\ \left|h_j(x)\right| \ \text{if } l+1 \le j \le p \end{cases} \tag{2}$$

Here, the absolute value of the equality constraint function ($\left|h_j(x)\right|$) can be treated as an inequality given by $G_j(x) = max\ \left\{0, \left|h_j(x)\right| - \delta\right\}$, where δ is a small positive value.

The general form of the penalty function ($p\ (x)$) and the corresponding evaluation function ($eval\ (x)$) can be described as follows [1]:

$$\begin{cases} p(x) = C\sum_{j=1}^{l+p} [G_j(x)]^{\beta} \\ eval\ (x) = \begin{cases} f(x) \text{ if } x \in F \\ f(x) + p(x) \text{ if } x \in U \end{cases} \end{cases} \tag{3}$$

where β and C are generalized dynamic or static coefficients, chosen according to the applied technique; F and U represent the feasible and infeasible spaces, respectively. When handling COPs, $p(x)$ is usually used to evaluate the infeasibility of the population.

3. Heat Transfer Search (HTS) Algorithm

The HTS algorithm is a relatively new population-based approach that belongs to the family of MHAs. It is inspired by the natural laws of thermodynamics and heat transfer; [18] declares that "any system usually attempts to attain an equilibrium state with the surroundings" [18]. It has been reported that the HTS algorithm mimics the thermal equilibrium behavior of the systems by considering three heat transfer phases, including the conduction phase, convection phase, and radiation phase [18]; each phase plays a crucial role in establishing the thermal equilibrium and attaining an equilibrium state. Similarly to other MHAs, this algorithm starts with a randomly initialized population, and the population is considered as a cluster of the system's molecules. These molecules aim to attain an equilibrium state with the surroundings through the three phases of heat transfer, by interacting with each other and with their surrounding environment. In the basic HTS algorithm, the population members are only updated through one phase of the three heat transfer phases in each iteration. The selection process for which of the three phases to be activated for updating the solutions in the particular iteration is carried out by a uniformly distributed random number R. This random number is generated in the range [0, 1], randomly, in each iteration to determine the phase that must be selected. In other words, the population members undergo the conduction phase when the random number R varies between 0 and 0.3333, the radiation phase when the random number R varies between 0.3333 and 0.6666, and the convection phase when the random number R varies between 0.6666 and 1. The greedy selection technique is the main selection mechanism for newly generated solutions in the HTS algorithm; this technique states that only new updated solutions which have a superior objective value will be accepted, and the solutions with an inferior objective value will be subsequently substituted by the best solutions. Hence, by comparing the difference between the current solution and the elite solutions, the greatest solution can be finally achieved. In the basic HTS algorithm, the main search process is performed by the elementary operations of the three heat transfer phases (conduction, convection, and radiation); the basic principle of each phase is briefly described in the subsequent subsections. The overall flow-chart of the original HTS method is illustrated in Figure 1.

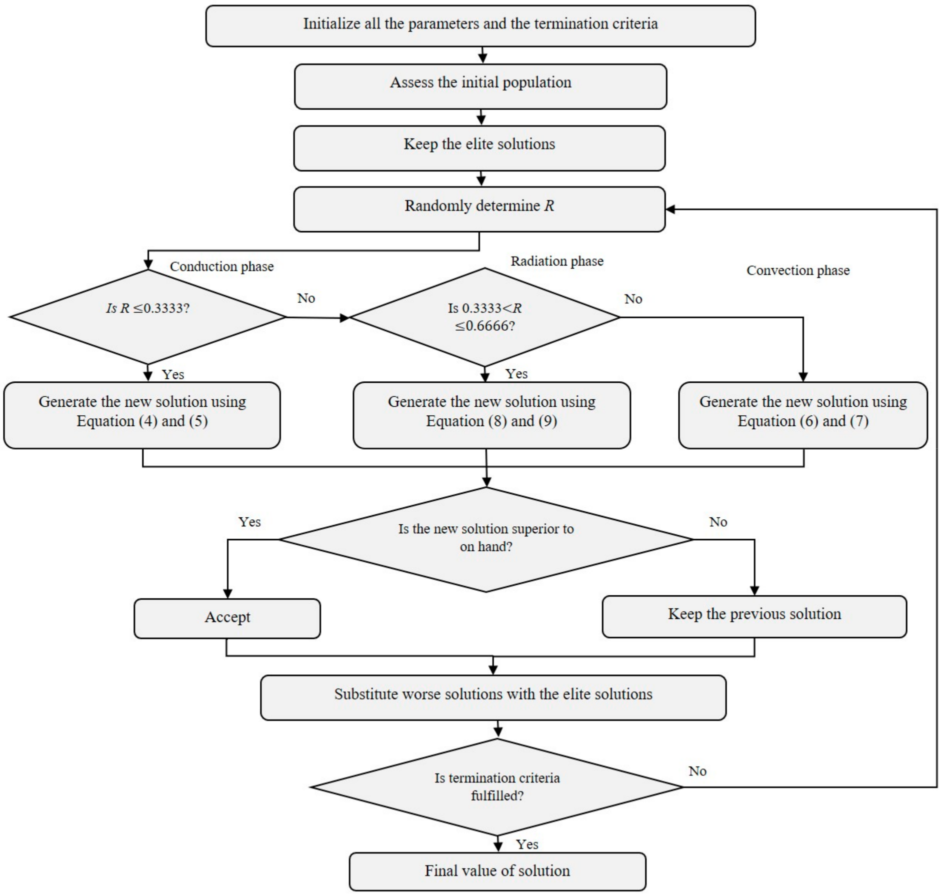
Initialize all the parameters and the termination criteria
Assess the initial population
Keep the elite solutions
Randomly determine R
Conduction phase
Radiation phase
Convection phase
Is R ≤0.3333?
No
Is 0.3333<R ≤0.6666?
No
Yes
Yes
Generate the new solution using Equation (4) and (5)
Generate the new solution using Equation (8) and (9)
Generate the new solution using Equation (6) and (7)
Yes
Is the new solution superior to on hand?
No
Accept
Keep the previous solution
Substitute worse solutions with the elite solutions
Is termination criteria fulfilled?
No
Yes
Final value of solution

Figure 1. The overall flowchart of the HTS algorithm.

3.1. Conduction Phase

The fundamental principle of the conduction phase in the original HTS algorithm is concerned with the influence of conduction among the substance's molecules, and the outcome is that the heat transfer takes place. Thus, by aiming to attain a state of thermal equilibrium through the conduction influence, the molecules that have higher energy transfer the heat to the molecules that have lower energy. In other words, the system here aims to neutralize the thermal imbalance among the substance's molecules. In the conduction phase of the HTS algorithm, a potential new solution is updated by utilizing the following formulas:

$$\left\{\begin{array}{l} X_{j,i}^{new} = X_{k,i}^{old} + CDS_1 \text{ If } f(X_j) > f(X_k) \\ X_{k,i}^{new} = X_{j,i}^{old} + CDS_2 \text{ If } f(X_k) > f(X_j) \\ CDS_1 = -R^2 X_{k,i}^{old} \\ CDS_2 = -R^2 X_{j,i}^{old} \end{array}\right\} if\ FE \le \frac{\max FEs}{CDF} \tag{4}$$

$$\left\{\begin{array}{l} X_{j,i}^{new} = X_{k,i}^{old} + CDS_3 \text{ If } f(X_j) > f(X_k) \\ X_{k,i}^{new} = X_{j,i}^{old} + CDS_4 \text{ If } f(X_k) > f(X_j) \\ CDS_3 = -r_i X_{k,i}^{old} \\ CDS_4 = -r_i X_{j,i}^{old} \end{array}\right\} if\ FE \ge \frac{\max FEs}{CDF} \tag{5}$$

where maxFEs denotes the maximum function evaluations; CDF indicates the conduction factor which is set to a value of 2 for balancing the exploration and exploitation in this phase; FE indicates the function evaluations; $X_{j,i}^{new}$ denotes the newly updated solution; $j = 1, 2, \ldots, n$; k indicates a randomly chosen solution; $j \neq k$, where $k \in (1, 2, \ldots, n)$; i represents a randomly chosen decision variable, and $i \in (1, 2, \ldots, m)$; R indicates the random variable that varies in the interval [0, 0.3333], and r_i indicates a random number that varies in the interval [0, 1]. Here, r_i and R^2 are in correlation with the conductance elements of the Fourier's law of heat conduction [18].

3.2. Convection Phase

The fundamental principle of the convection phase in the basic HTS algorithm is that due to the convection effect between the surrounding temperature and the system, heat transfer occurs. Thus, by attaining a state of thermal equilibrium through the convection effect, the mean temperature of the system interacts with the adjacent surrounding temperature to neutralize the thermal imbalance. In the convection phase, the mean temperature of the system is represented by the mean of the population members, which is marked as X_{ms}, whereas the surrounding temperature is represented by the best solution, which is marked as X_s. Therefore, a new solution is updated in this part of the algorithm using the following equation:

$$\left\{\begin{array}{l} X_{j,i}^{new} = X_{j,i}^{old} + \text{COS} \\ \text{COS} = R(X_s - X_{ms})\text{TCF} \end{array}\right\} \tag{6}$$

$$TCF = \left\{\begin{array}{l} abs(R - r_i) \text{ If } FE \le \max FEs/COF \\ round(1 + r_i) \text{ If } FE \ge \max FEs/COF \end{array}\right. \tag{7}$$

where COF represents the convection factor, and is set to a value of 10 for balancing the exploration and exploitation in this phase; FE denotes the function evaluations; TCF indicates the temperature change factor; $X_{j,i}^{new}$ denotes the new updated solution; $j = 1, 2, \ldots, n$; $i = 1, 2, \ldots, m$; r_i is a random number which varies in the interval [0, 1], and R is the random variable that varies in the interval [0.6666, 1]. Here, r_i and R are in correlation with the convection elements of Newton's law of cooling [18]. The value of TCF varies randomly in the range [0, 1] in the first part of this phase, whereas it varies either as 1 or 2 in the second part of this phase.

3.3. Radiation Phase

The fundamental principle of the radiation phase in the original HTS algorithm is that the system attempts to neutralize the thermal imbalance by interacting within the system itself (for example, the other solution), or with the surrounding temperature (for example, the best solution). Here, the system aims to establish a state of thermal balance. In the radiation phase of the HTS algorithm, a potential new solution is updated by utilizing the subsequent formulas:

$$\left\{\begin{array}{l} \mathrm{X}_{j,i}^{new} = \mathrm{X}_{j,i}^{old} + \mathrm{RDS}_1 \text{ If } f(X_j) > f(X_k) \\ \mathrm{X}_{k,i}^{new} = \mathrm{X}_{k,i}^{old} + \mathrm{RDS}_2 \text{ If } f(X_k) > f(X_j) \\ \mathrm{RDS}_1 = R\left(\mathrm{X}_{k,i}^{old} - \mathrm{X}_{j,i}^{old}\right) \\ \mathrm{RDS}_2 = R\left(\mathrm{X}_{j,i}^{old} - \mathrm{X}_{k,i}^{old}\right) \end{array}\right\} if\ FE \leq \frac{\mathrm{max}FEs}{RDF} \tag{8}$$

$$\left\{\begin{array}{l} \mathrm{X}_{j,i}^{new} = \mathrm{X}_{j,i}^{old} + \mathrm{RDS}_3\ if\ f(X_j) > f(X_k) \\ \mathrm{X}_{k,i}^{new} = \mathrm{X}_{k,i}^{old} + \mathrm{RDS}_4\ if\ f(X_k) > f(X_j) \\ \mathrm{RDS}_3 = r_i\left(\mathrm{X}_{k,i}^{old} - \mathrm{X}_{j,i}^{old}\right) \\ \mathrm{RDS}_4 = r_i\left(\mathrm{X}_{j,i}^{old} - \mathrm{X}_{k,i}^{old}\right) \end{array}\right\} if\ FE \geq \frac{\mathrm{max}FEs}{RDF} \tag{9}$$

where RDF denotes the radiation factor whose value is set at 2 for balancing the exploitation and exploration in this phase; FE indicates the function evaluations; $X_{j,i}^{new}$ denotes the new updated solution; $j = 1, 2, \ldots, n$; $i = 1, 2, \ldots, m$; k represents the randomly selected solutions; $j \neq k$, and $k \in (1, 2, \ldots, n)$; r_i indicates a random number which varies in the interval [0, 1], and R denotes the random variable which varies in the interval [0.3333, 0.6666]. Here, r_i and R are in correlation with the radiation parameters of the Stefan–Boltzmann law [18].

4. Proposed MHTS–TR Algorithm

In this section, by integrating the tandem running (TR) strategy into the basic HTS algorithm, a novel variant of the HTS algorithm named MHTS–TR is presented. The proposed variant is described in detail in the following subsections.

4.1. Tandem-Running Strategy

In nature, animals acquire information socially, and comprehensively utilize such information to allow them to exist in large, collaborative societies. The behavior of energetically sharing personal information with other members of a social group is referred to as cooperative communication [27]. Tandem running (TR) is a form of recruitment seen in some groups of ants, in which one member guides another to a crucial resource, such as a superior nest location. For instance, the Temnothorax Albipennis ant species use tandem-run recruitment, where one ant acts as a leader and guides a single follower from the nest to food; thus, the follower finds food more rapidly than when seeking alone. The tandem leader is familiar with the location of the food resource whereas the follower is naive. The course and speed of the leader are governed by the follower, who taps the leader's legs and gaster with its antennae [28]. The following ant acquires knowledge regarding the placemarks and surroundings en route, and thus learns to take a more linear route, usually returning to its nest quicker than before [28]. Generally speaking, when ant colonies are suddenly placed in a complex and unfamiliar environment, this information sharing behavior commonly appears. Thus, this behavior can also be mimicked in complex environments for exploring useful formation.

4.2. Outline of the MHTS–TR Algorithm

Inspired by the cooperative exploration behavior of the Temnothorax Albipennis ant species in unknown areas, described as a "move-to-improve" method, this strategy is incorporated into the HTS algorithm, and a new variant named MHTS–TR has been proposed. The overall scheme of the MHTS–TR algorithm is shown in Figure 2. For this approach, we divided the population members into two categories (feasible and infeasible) at each iteration, depending on constraint violation. The members that did not violate the constraints were termed as "leaders", and were evolved within the feasible region by the HTS algorithm. By contrast, the violated members were termed as "followers", and were further classified into two parts (X_{HV} and X_{SV}) depending on their violation degree. X_{HV} represented the member with a higher degree of violation, with a position often further away from the feasible region. Thus, a member was randomly selected from the feasible region to be the leader, and the followers then moved toward the neighborhood of this leader to search within the feasible region. As a result, the members within the infeasible region that did not contribute to the population were moved toward the feasible region. Meanwhile, as each follower randomly selected its leader, the population density within the feasible region increased evenly and, hence, increased the diversity of the population within the feasible region. On another hand, X_{SV} represented the member with a relatively lower degree of violation, which was considered as it was almost close to the feasible region. It selected the nearest member that was located in the feasible region to be its leader, and moved towards it; hence, the boundaries of the feasible region were gradually searched by approximating towards the leader. In this way, the members with infeasible information nearby the boundaries were utilized to explore the superior areas that were hidden nearby the boundaries of the feasible region.

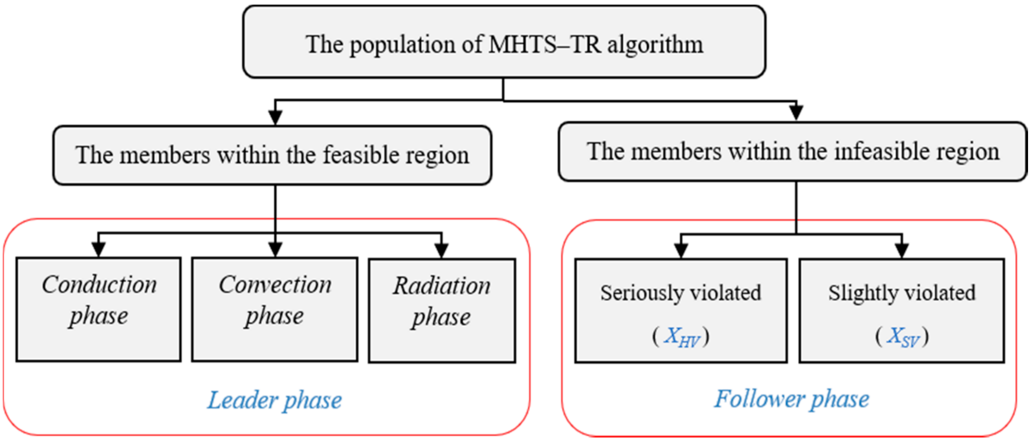
The population of MHTS–TR algorithm
The members within the feasible region
The members within the infeasible region
Conduction phase
Convection phase
Radiation phase
Seriously violated (X_{HV})
Slightly violated (X_{SV})
Leader phase
Follower phase

Figure 2. The overall scheme of the MHTS–TR algorithm.

Therefore, due to the methods used by X_{HV} and X_{SV} to select their respective leaders being carried out through random selection and distance judgment, which was irrespective of the fitness value, there was no issue of the members being overly concentrated around the global optimal member. Thus, the non-connectivity in the feasible region did not affect the distribution of the population in each feasible region. By alternating between these two complementary phases, the MHTS–TR method was expected to explore various zones of the search space without being easily trapped in a local optimum. The moving strategies of X_{HV} and X_{SV} are shown in Figure 3.

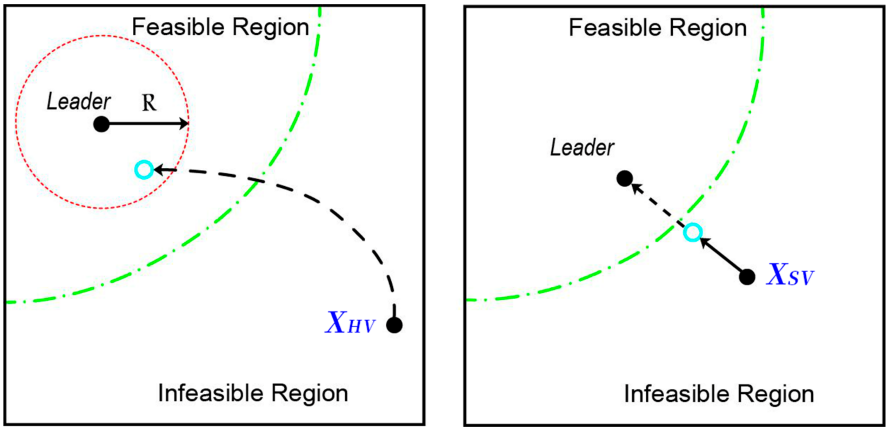

Figure 3. The moving strategies of X_{HV} and X_{SV}.

4.3. The Overall Process of MHTS–TR Method

Firstly, we assumed that the population M was the number of members that searched in an n-dimensional space ($S \in R^n$), and $\Omega \subseteq S$ was the feasible region of the solution space. At the beginning of the kth iteration, the distance matrix ($Dis^k = \left[{Dis_1}^k, \ldots, {Dis_i}^k, \ldots, {Dis_M}^k\right]$) for all the members was calculated, in which ${Dis_i}^k$ was an m-dimensional vector that represented the distance between the member i and other members, and ${Dis_i}^k = \left[{dis_{i1}}^k, \ldots, {dis_{ij}}^k, \ldots, {dis_{iM}}^k\right]$, where ${dis_{ij}}^k$ was the Euclidean distance between the member i and member j ($1 \leq j \leq M$ and $j \neq i$).

Secondly, all members of the population were evaluated as to whether they were within the feasible region or not, and then divided into two sub-populations, accordingly. We assumed that Fb was the number of members within the infeasible region, where $0 \leq Fb_k \leq M$, and the number of members within the feasible region was $M - Fb_k$. These members were considered as "leaders", and would evolve according to the basic rules of the conduction, convection, and radiation phases of the HTS algorithm.

The degree of constraint violations for all the members in the infeasible region was calculated $G(x) = [G(x_1), \ldots, G(x_i), \ldots, G(x_{Fb})]$, i = 1, 2, … , Fb, and then $G(x)$ was sorted in descending order. A number of members with the highest degree of violation (denoted by SN) were selected to be X_{HV}, and the other members were selected to be X_{SV}. Here, the SN was jointly determined by the number of members within the infeasible region (Fb) and the X_{HV} ratio (ps_k) at the kth iteration. It can be calculated as follows:

$$SN = \lfloor Fb * ps_k \rfloor \tag{10}$$

$$ps_k = ps_{\min} + \frac{(ps_{\max} - ps_{\min}) * k}{MaxIter} \tag{11}$$

where the operational sign "$\lfloor \ \rfloor$" means round number; $ps_{\min}$ and $ps_{\max}$ are the initial and terminal values of ps, respectively, where $0 \leq ps_{\min} < ps_{\max} \leq 1$. The value of ps_k increased with the increase in iterations throughout the whole process of calculation, meaning that the MHTS–TR algorithm had a high tolerance level for the violated members at the beginning of the calculation. In other words, most of the members within the infeasible region were moved towards the feasible region using the approximation manner, whereas in the later stage of the calculation, the tolerance of violated members was gradually tightened by the increasing the ratio of X_{HV}. Most of them were directly updated into the interior of the feasible region for searching. The detailed evolutionary mechanisms of the X_{HV} and X_{SV} are given as follows:

(1) The moving strategy of X_{HV}:

We assume that, a member i is selected to be X_{HV} at the kth iteration, when its position is $x_i^k = (x_{i1}^k, x_{i2}^k, \cdots x_{in}^k) \notin \Omega$, then a member j will be randomly selected from the feasible region to be the leader of this member, so that the member i will be updated to a new position around the member j, where the position of the member j is $x_j^k = (x_{j1}^k, x_{j2}^k, \cdots x_{in}^k) \in \Omega$.

Firstly, the leader (member j) will select the closest member g which is also within the feasible region according to the distance ${Dis_j}^k$, when its position is $x_g^k = (x_{g1}^k, x_{g2}^k, \cdots x_{gn}^k) \in \Omega$, and then the Euclidian distance between the member j and member g on each dimension is calculated ${NDis_j}^k = [edis_1, \ldots, edis_h, \ldots edis_n]$, where $edis_h$ is the distance between the member j and member g on the h dimension, and its calculation formula is as below:

$$edis_h = \left| x_{gh}^k - x_{jh}^k \right| \tag{12}$$

Then, the X_{HV} (member i) will update its position using the subsequent formula.

$$x_i^{k+1} = x_j^k + rand(n) \times NDis_j^k \tag{13}$$

where $rand(n)$ is an n-dimensional random vector, it is uniformly distributed between 0 and 1, and the operator "×" means calculating the element-wise product of the two vectors.

(2) The moving strategy of X_{SV}:

We assume that, at the kth iteration, when a member i is selected to be X_{SV}, it selects the nearest member j within the feasible region depending on the distance ${Dis_i}^k$ between this member and other members, and then approximates towards the member j. Thus, it will update its position as follows:

$$x_i^{k+1} = x_i^k + c * rand(n) * (x_j^k - x_i^k) \tag{14}$$

where c is the velocity factor of X_{SV}, it is used to adjust the velocity of the X_{SV} to approximate toward the feasible region.

The overall process of this approach is illustrated by the flowchart diagram in Figure 4.

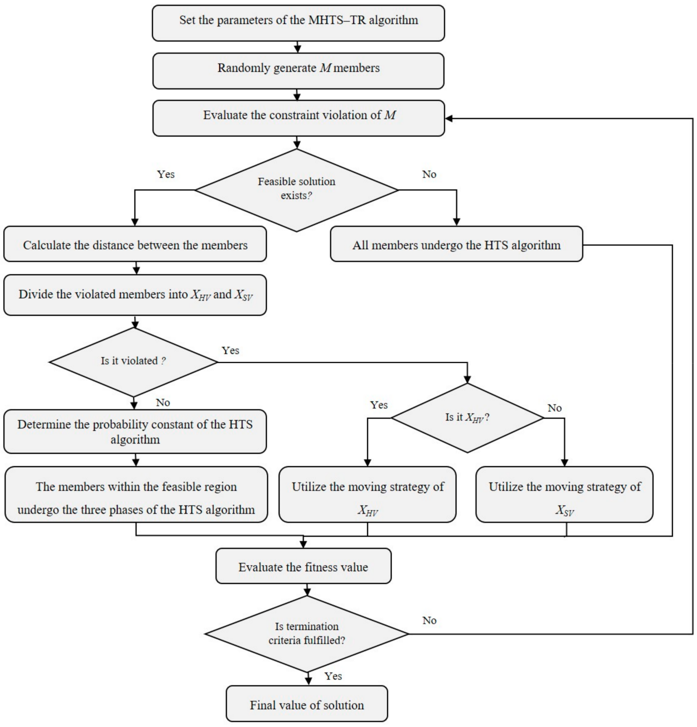
Set the parameters of the MHTS–TR algorithm
Randomly generate M members
Evaluate the constraint violation of M
Yes
Feasible solution exists?
No
Calculate the distance between the members
All members undergo the HTS algorithm
Divide the violated members into X_{HV} and X_{SV}
Is it violated ?
Yes
No
Determine the probability constant of the HTS algorithm
Yes
Is it X_{HV}?
No
The members within the feasible region undergo the three phases of the HTS algorithm
Utilize the moving strategy of X_{HV}
Utilize the moving strategy of X_{SV}
Evaluate the fitness value
Is termination criteria fulfilled?
No
Yes
Final value of solution

Figure 4. Flowchart of the MHTS–TR method.

5. Numerical Experiments and Discussion

In this section, we outline how the overall effectiveness of the MHTS–TR approach was verified by a set of 24 well-defined COPs of Congress on Evolutionary Computation

2006 (CEC 2006) [29–31]. Moreover, comparisons between the new variant and several other well-established MHAs, such as differential evolution (DE), particle swarm optimization (PSO), biogeography-based optimization (BBO), artificial bee colony (ABC), teaching-learning-based optimization (TLBO), and the original heat transfer search (HTS) algorithm were conducted. These comparative approaches were examined against the considered benchmark problems noted previously in the literature [18]. Therefore, they were employed for comparison with the proposed variant; this is notable since a common experimental platform is required to make fair comparisons against the competitor algorithms. Hence, the population size (*NP*) was set at 50, and the maximum number of function evaluations (*maxIter*) was set to 240,000. Furthermore, the computational results obtained from 100 independent runs, such as the best value (Best), mean value (Mean), worst value (Worst), standard deviations (Std), and success rate (SR) were listed. To further quantify the performances of the competitors statistically, a Friedman rank test was also conducted on the mean and best solutions obtained by the comparative approaches. The constraint-handling technique used with the considered competitors was the static penalty method.

5.1. Benchmark Functions

The considered test functions offered challenges to the MHAs and can be classified into different types: non-linear, linear, quadratic, polynomial, and cubic functions. The detailed characteristics of each test function are illustrated in Table 1, and the detailed mathematical formulation of each test function with its constraints can be found in the literature [18].

Table 1. The characteristics of 24 COPs used in the investigation. n denotes the number of decision variables; F indicates the global optimum value; P (%) indicates the estimated ratio between the search space and the feasible domain; NI represents the number of non-linear inequality constraints; LI denotes the number of linear inequality constraints; NE denotes the number of non-linear equality constraints; LE indicates the number of linear equality constraints, and *ac* represents the number of active constraints at the optimum solution.

Function	Type of Function	n	F	P (%)	NI	LI	NE	LE	ac
C01	Quadratic	13	−15.0000	0.0111	0	9	0	0	6
C02	Non linear	20	-8.036×10^{-1}	99.9971	2	0	0	0	1
C03	Polynomial	10	−1.0005	0.0000	0	0	1	0	1
C04	Quadratic	5	−30,665.5	52.1230	6	0	0	0	2
C05	Cubic	4	5126.496	0.0000	0	2	3	0	3
C06	Cubic	2	−6961.81	0.0066	2	0	0	0	2
C07	Quadratic	10	24.3062	0.0003	5	3	0	0	6
C08	Non linear	2	-9.582×10^{-2}	0.8560	2	0	0	0	0
C09	Polynomial	7	680.63	0.5121	4	0	0	0	2
C10	Linear	8	7049.248	0.0010	3	3	0	0	6
C11	Quadratic	2	7.499×10^{-1}	0.0000	0	0	1	0	1
C12	Quadratic	3	−1.0000	4.7713	1	0	0	0	0
C13	Non linear	5	5.390×10^{-2}	0.0000	0	0	3	0	3
C14	Non linear	10	−47.7649	0.0000	0	0	0	3	3
C15	Quadratic	3	961.715	0.0000	0	0	1	1	2
C16	Non linear	5	−1.9051	0.0204	34	4	0	0	4
C17	Non linear	6	8853.539	0.0000	0	0	4	0	4
C18	Quadratic	9	-8.6602×10^{-1}	0.0000	13	0	0	0	6
C19	Non linear	15	32.6556	33.4761	5	0	0	0	0
C20	Linear	24	2.0498×10^{-1}	0.0000	6	0	12	2	16
C21	Linear	7	193.7245	0.0000	1	0	5	0	6
C22	Linear	22	236.4309	0.0000	1	0	11	8	19
C23	Linear	9	−400.055	0.0000	2	0	1	3	6
C24	Linear	2	−5.50801	79.6556	2	0	0	0	2

5.2. Simulation Results and Analysis

The comparative results of each approach on C01 to C13 are displayed in Table 2. It can be observed from the results table that the MHTS–TR, HTS, and ABC algorithms markedly outperformed the other four algorithms on the results of the quadratic function (C01). Moreover, they produced a 100% success rate compared to the other algorithms on this function. In terms of the non-linear function (C02), although the ABC method produced superior results to the other approaches, all comparative results were considered, as they failed to attain the global optimum. For the polynomial test function (C03), the ABC approach produced superior results than other methods; however, the MHTS–TR and HTS methods obtained competitive performances, and achieved the global optimum on this test function. On the quadratic function (C04), all approaches produced equally great results except for the BBO algorithm. On the cubic function (C05), the MHTS–TR, HTS, and TLBO methods showed competitive results over the other methods, and the mean value acquired by the MHTS–TR was superior to that of the HTS and TLBO, respectively. On the cubic function (C06), the MHTS–TR, ABC, TLBO, HTS, and PSO approaches produced equally great values. On the quadratic function (C07), the mean value obtained by the MHTS–TR, HTS, ABC, and DE algorithms was better than those of the other competitive algorithms. On the non-linear function (C08) and polynomial function (C09), all the competitive methods (except the BBO) produced equally good results. On the linear function (C10), the mean value acquired by the MHTS–TR was superior to those of the other competitive algorithms. All the competitive algorithms (except the DE and BBO) produced equally great results on the quadratic function (C11). On the quadratic function (C12), all approaches (except the PSO) produced equally good solutions. All the competitive approaches failed to attain the global optimum value on the non-linear function (C13); however, the mean value obtained by the MHTS–TR and PSO approaches was better than those of other competitive algorithms.

Table 2. Computational results of C01–C13 for the MHTS–TR method with other MHAs. (The experimental results of the competitor algorithms are derived from the literature [18]).

Function		DE	PSO	BBO	ABC	TLBO	HTS	MHTS–TR
C01	Best	−15.0000	−15.0000	−14.977	−15.0000	−15.0000	−15.0000	−15.0000
	Mean	−14.555	−14.71	−14.7698	−15.0000	−10.782	−15.0000	−15.0000
	Worst	−11.828	−13.0000	−14.5882	−15.0000	−6.0000	−15.0000	−15.0000
	SR	94	38	0	100	26	100	100
C02	Best	-4.720×10^{-1}	-6.691×10^{-1}	-7.821×10^{-1}	-8.035×10^{-1}	-7.835×10^{-1}	-7.517×10^{-1}	-7.689×10^{-1}
	Mean	-6.650×10^{-1}	-4.199×10^{-1}	-7.642×10^{-1}	-7.924×10^{-1}	-6.705×10^{-1}	-6.437×10^{-1}	-7.678×10^{-1}
	Worst	-4.720×10^{-1}	-2.994×10^{-1}	-7.389×10^{-1}	-7.497×10^{-1}	-5.518×10^{-1}	-5.482×10^{-1}	-7.662×10^{-1}
	SR	0	0	0	0	0	0	0
C03	Best	-9.939×10^{-1}	−1	−1.0005	−1.0000	−1.0005	−1.0005	−1.0005
	Mean	−1.0000	7.648×10^{-1}	-3.957×10^{-1}	−1.0000	-8.000×10^{-1}	-9.004×10^{-1}	-9.601×10^{-1}
	Worst	−1.00000	-4.640×10^{-1}	-4.550×10^{-2}	−1.0000	0.0000	0.0000	0.0000
	SR	41	59	23	67	74	86	90
C04	Best	−30,665.539	−30,665.539	−30,665.539	−30,665.539	−30,665.5387	−30,665.5387	−30,665.5387
	Mean	−30,665.539	−30,665.539	−30,411.865	−30,665.539	−30,665.5387	−30,665.5387	−30,665.5387
	Worst	−30,665.539	−30,665.539	−29,942.3	−30,665.539	−30,665.5387	−30,665.5387	−30,665.5387
	SR	100	100	16	100	100	100	100
C05	Best	5126.484	5126.484	5134.2749	5126.484	5126.486	5126.486	5126.487
	Mean	5264.27	5135.973	6130.5289	5185.714	5126.6184	5126.5152	5126.5030
	Worst	5534.61	5249.825	7899.2756	5438.387	5127.714	5126.6831	5126.5049
	SR	93	61	0	28	92	95	97

Table 2. *Cont.*

Function		DE	PSO	BBO	ABC	TLBO	HTS	MHTS–TR
C06	Best	−6954.434	−6961.814	−6961.8139	−6961.814	−6961.814	−6961.814	−6961.814
	Mean	−6954.434	−6961.814	−6181.7461	−6961.813	−6961.814	−6961.814	−6961.814
	Worst	−6954.434	−6961.814	−5404.4941	−6961.805	−6961.814	−6961.814	−6961.814
	SR	100	100	21	100	100	100	100
C07	Best	24.306	24.37	25.6645	24.33	24.3101	24.3104	24.3086
	Mean	24.31	32.407	29.829	24.473	24.837	24.4945	24.4879
	Worst	24.33	56.055	37.6912	25.19	27.6106	25.0083	24.9563
	SR	26	21	0	28	23	37	41
C08	Best	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}
	Mean	-9.5825×10^{-2}	-9.5825×10^{-2}	-9.5824×10^{-2}	-9.5825×10^{-2}	-9.5825×10^{-2}	-9.5825×10^{-2}	-9.5824×10^{-2}
	Worst	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}	-9.582×10^{-2}
	SR	100	100	94	100	100	100	100
C09	Best	680.63	680.63	680.6301	680.634	680.6301	680.6301	680.6301
	Mean	680.63	680.63	692.7162	680.64	680.6336	680.6329	680.6326
	Worst	680.631	680.631	721.0795	680.653	680.6456	680.644	680.642
	SR	95	84	26	89	91	96	98
C10	Best	7049.548	7049.481	7679.0681	7053.904	7250.9704	7049.4836	7049.4833
	Mean	7147.334	7205.5	8764.9864	7224.407	7257.0927	7119.7015	7050.1982
	Worst	9264.886	7894.812	9570.5714	7604.132	7291.3779	7252.0546	7128.9652
	SR	0	0	0	0	0	0	94
C11	Best	7.520×10^{-1}	7.490×10^{-1}	7.499×10^{-1}	7.500×10^{-1}	7.499×10^{-1}	7.499×10^{-1}	7.499×10^{-1}
	Mean	9.010×10^{-1}	7.490×10^{-1}	8.305×10^{-1}	7.500×10^{-1}	7.499×10^{-1}	7.499×10^{-1}	7.499×10^{-1}
	Worst	1.00	7.490×10^{-1}	9.289×10^{-1}	7.500×10^{-1}	7.499×10^{-1}	7.499×10^{-1}	7.499×10^{-1}
	SR	19	100	57	100	100	100	100
C12	Best	−1.00	−1.00	−1.00	−1.00	−1.00	−1.00	−1.00
	Mean	−1.00	9.988×10^{-1}	−1.00	−1.00	−1.00	−1.00	−1.00
	Worst	−1.00	-9.940×10^{-1}	−1.00	−1.00	−1.00	−1.00	−1.00
	SR	100	100	100	100	100	100	100
C13	Best	3.850×10^{-1}	8.565×10^{-2}	6.282×10^{-1}	7.600×10^{-1}	4.401×10^{-1}	3.731×10^{-1}	4.679×10^{-1}
	Mean	8.720×10^{-1}	5.693×10^{-1}	1.09289	9.680×10^{-1}	6.905×10^{-1}	6.694×10^{-1}	5.841×10^{-1}
	Worst	9.900×10^{-1}	1.793361	1.45492	1.00	9.560×10^{-1}	7.975×10^{-1}	6.416×10^{-1}
	SR	0	0	0	0	0	0	0

Table 3 shows the Friedman rank test, which was conducted on the mean and best solutions obtained by the comparative methods on the functions C01 to C13. It can be seen from the result table that the MHTS–TR algorithm ranked first to obtain the mean solution, followed by the original HTS, ABC, and the rest of the other approaches, whereas it secured the second rank after the PSO algorithm to obtain the best solution on these test functions.

The results of the comparative algorithms on the functions C14 to C24 are shown in Table 4. It can be observed from the results table that all the methods failed to attain the global optimum value on the non-linear function (C14), which was a difficult function. However, the MHTS–TR algorithm produced a superior mean value compared to the other approaches. The mean result obtained by the MHTS–TR and DE algorithms outperformed the other algorithms on the quadratic function (C15). On the non-linear function (C16), it can be seen that all approaches (except the BBO) showed identical results and have succeeded to attain the global optimum value. On the non-linear functions (C17 and C19), and the quadratic function (C18), the mean results obtained by the TLBO, DE, and ABC algorithms were superior to the results of the other comparative algorithms. However, the MHTS–TR algorithm generated better results than the original HTS on these functions. On the rest of the linear functions (C20 to C24), due to C20, C22 and C23 having test problems as difficult functions, it can be seen that all the approaches failed to attain the global optimum value. However, the MHTS–TR method generated superior mean results compared to the other comparative approaches on the functions C20 and C23. The mean result and success rate obtained by the MHTS–TR algorithm on the function C21 was better than those of the other algorithms. On the function C24, it can be observed that all algorithms attained the global optimum value, and also shown identical results.

Table 3. Friedman rank of C01–C13 for the solutions obtained by the MHTS–TR and other MHAs. (The experimental results of the competitors are derived from the literature [18]).

The Mean Solution				The Best Solution			
Method	**Friedman Value**	**Normalized Value**	**Rank**	**Method**	**Friedman Value**	**Normalized Value**	**Rank**
DE	53.5	1.53	6	DE	50.5	1.23	6
PSO	47	1.34	4	PSO	41	1	1
BBO	84	2.4	7	BBO	63.5	1.55	7
ABC	38	1.09	3	ABC	47	1.15	4
TLBO	52	1.49	5	TLBO	47.5	1.16	5
HTS	35	1	2	HTS	45.5	1.11	3
MHTS–TR	34.2	0.93	1	MHTS–TR	43.8	1.04	2

Table 4. Computational results of C14–C24 for the MHTS–TR method with other MHAs. (The experimental results of the competitor algorithms are derived from the literature [18]).

Function		DE	PSO	ABC	BBO	TLBO	HTS	MHTS–TR
C14	Best	−45.7372	−44.9343	−44.6431	54.6979	−46.5903	−47.7278	−47.7391
	Mean	−29.2187	−40.8710	−40.1071	175.9832	−39.9725	−46.4076	−46.6872
	STD	1.36×10^{1}	2.2900	7.1400	7.90×10^{1}	1.15×10^{1}	8.53×10^{-1}	7.29×10^{-1}
	Worst	−12.7618	−37.5000	−23.3210	257.7061	−17.4780	−45.0648	−46.4135
	SR	0	0	0	0	0	0	0
C15	Best	961.7150	961.7150	961.7568	962.6640	961.7150	961.7150	961.7150
	Mean	961.7537	965.5154	966.2868	1001.4367	962.8641	961.7500	961.7483
	STD	1.22×10^{-1}	3.7200	3.1200	4.74×10^{1}	1.4900	1.11×10^{-1}	1.18×10^{-1}
	Worst	962.1022	972.3170	970.3170	1087.3557	964.8922	962.0653	961.9642
	SR	73	53	42	0	81	83	84
C16	Best	−1.9052	−1.9052	−1.9052	−1.9052	−1.9052	−1.9052	−1.9052
	Mean	−1.9052	−1.9052	−1.9052	−1.6121	−1.9052	−1.9052	−1.9052
	STD	2.34×10^{-16}	2.34×10^{-16}	2.34×10^{-16}	2.58×10^{-1}	2.34×10^{-16}	2.34×10^{-16}	2.34×10^{-16}
	Worst	−1.9052	−1.9052	−1.9052	−1.1586	−1.9052	−1.9052	−1.9052
	SR	100	100	100	18	100	100	100
C17	Best	8854.6501	8857.5140	8859.7130	9008.5594	8853.5396	8853.5396	8853.5396
	Mean	8932.0444	8899.4721	8941.9245	9384.2680	8876.5071	8877.9175	8876.7492
	STD	4.68×10^{1}	3.79×10^{1}	4.26×10^{1}	3.06×10^{2}	3.02×10^{1}	3.09×10^{1}	3.01×10^{1}
	Worst	8996.3215	8965.4010	8997.1450	9916.7742	8919.6595	8932.0712	8917.3041
	SR	0	0	0	0	58	26	29
C18	Best	-8.653×10^{-1}	-8.660×10^{-1}	-8.660×10^{-1}	-6.573×10^{-1}	-8.660×10^{-1}	-8.660×10^{-1}	-8.660×10^{-1}
	Mean	-8.616×10^{-1}	-8.276×10^{-1}	-8.658×10^{-1}	-5.681×10^{-1}	-8.656×10^{-1}	-7.703×10^{-1}	-8.512×10^{-1}
	STD	3.67×10^{-3}	1.11×10^{-1}	3.37×10^{-4}	8.55×10^{-2}	9.67×10^{-4}	1.01×10^{-1}	1.16×10^{-1}
	Worst	-8.551×10^{-1}	-5.108×10^{-1}	-8.652×10^{-1}	-3.887×10^{-1}	-8.629×10^{-1}	-6.746×10^{-1}	-8.423×10^{-1}
	SR	61	56	73	0	64	47	53

Table 4. *Cont.*

Function		DE	PSO	ABC	BBO	TLBO	HTS	MHTS–TR
C19	Best	32.6851	33.5358	33.3325	39.1471	32.7916	32.7132	32.7085
	Mean	32.7680	36.8443	36.0078	51.8769	34.0792	32.7903	32.7696
	STD	6.28×10^{-2}	2.0400	1.8300	1.12×10^{1}	9.33×10^{-1}	1.53×10^{-1}	1.78×10^{-1}
	Worst	32.9078	39.8443	38.5614	71.3106	36.1935	33.2140	32.932
	SR	0	0	0	0	0	0	0
C20	Best	2.4743×10^{-1}	2.474×10^{-1}	2.4743×10^{-1}	1.26181	2.4743×10^{-1}	2.4743×10^{-1}	2.4743×10^{-1}
	Mean	2.616×10^{-1}	9.723×10^{-1}	8.053×10^{-1}	1.43488	1.22037	2.5519×10^{-1}	2.4817×10^{-1}
	STD	1.91×10^{-2}	6.34×10^{-1}	5.93×10^{-1}	2.20×10^{-1}	5.89×10^{-1}	1.25×10^{-2}	1.61×10^{-2}
	Worst	2.876×10^{-1}	1.87320	1.52017	1.98625	1.84773	2.733×10^{-1}	2.596×10^{-1}
	SR	0	0	0	0	0	0	0
C21	Best	193.7346	193.7311	193.7343	198.8151	193.7246	193.7264	193.7253
	Mean	366.9193	345.6595	275.5436	367.2513	264.6092	256.6091	252.3172
	STD	9.13×10^{1}	6.36×10^{1}	6.05×10^{1}	1.34×10^{2}	9.23×10^{1}	6.63×10^{1}	6.72×10^{1}
	Worst	418.4616	409.1320	330.1638	581.2178	393.8295	320.2342	313.0856
	SR	24	12	36	0	35	48	51
C22	Best	1.25×10^{18}	1.68×10^{22}	2.82×10^{8}	1.02×10^{15}	4.50×10^{17}	2.16×10^{3}	1.23×10^{3}
	Mean	1.78×10^{19}	1.63×10^{23}	4.10×10^{17}	1.41×10^{16}	1.61×10^{19}	1.36×10^{6}	1.18×10^{6}
	STD	1.17×10^{19}	9.17×10^{22}	4.72×10^{17}	1.96×10^{16}	1.51×10^{19}	4.20×10^{6}	3.87×10^{6}
	Worst	2.67×10^{19}	3.25×10^{25}	1.25×10^{18}	6.70×10^{16}	4.06×10^{19}	1.33×10^{7}	1.22×10^{7}
	SR	0	0	0	0	0	0	0
C23	Best	−72.6420	−105.9826	−43.2541	2.3613	−385.0043	−390.6472	−394.0927
	Mean	−7.2642	−25.9179	−4.3254	22.1401	−83.7728	−131.2522	−134.8962
	STD	2.30×10^{1}	4.30×10^{1}	1.37×10^{1}	2.51×10^{1}	1.59×10^{2}	1.67×10^{2}	1.62×10^{2}
	Worst	0.00	0.00	0.00	74.6089	0.00	0.00	0.00
	SR	0	0	0	0	0	0	0
C24	Best	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080	−5.5080
	Mean	−5.5080	−5.5080	−5.5080	−5.4982	−5.5080	−5.5080	−5.5080
	STD	9.36×10^{-16}	9.36×10^{-16}	9.36×10^{-16}	6.75×10^{-3}	9.36×10^{-16}	9.36×10^{-16}	9.36×10^{-16}
	Worst	−5.5080	−5.5080	−5.5080	−5.4857	−5.5080	−5.5080	−5.5080
	SR	100	100	100	27	100	100	100

Table 5 shows the Friedman rank test, which was conducted on the mean and best solutions acquired by the comparative methods on the functions C14 to C24. It can be seen from the results table that the MHTS–TR algorithm ranked first to obtain the mean and best solutions, followed by the original HTS, TLBO, DE, and the other algorithms.

Table 5. Friedman rank of C14–C24 for the solutions obtained by the MHTS–TR and other MHAs. (The experimental results of the competitors are derived from the literature [18]).

The Mean Solution				The Best Solution			
Method	Friedman Value	Normalized Value	Rank	Method	Friedman Value	Normalized Value	Rank
DE	37	1.76	4	DE	38.5	1.64	4
PSO	41	1.95	6	PSO	40	1.7	5
BBO	38	1.81	5	BBO	42.5	1.81	6
ABC	62	2.95	7	ABC	58	2.47	7
TLBO	32	1.52	3	TLBO	28.5	1.21	3
HTS	21	1	2	HTS	23.5	1	2
MHTS–TR	20.2	0.91	1	MHTS–TR	22.3	0.93	1

Due to the appropriate trade-off between exploitation and exploration, our method was carried out by integrating different search mechanisms. The proposed method could alleviate the probability of local optima stagnation, and finally converge to a global feasible solution. Hence, it achieved the global optimum solution for more test problems compared to the original HTS method and the rest of the other competitive methods.

5.3. Further Discussion

To further analyze the advantages of the MHTS–TR algorithm in handling COPs, a comparison between the basic HTS and MHTS–TR on the test problem C10 was conducted. This comparison was carried out by counting the number of feasible solutions during the optimization course. To facilitate the comparison, the population size of both approaches was set at 100, and the maximum number of iterations was set to 1000.

Figure 5 shows the number of feasible solutions obtained by both approaches in 1000 iterations. It can be seen that both algorithms found feasible solutions in the 10th iteration, and the number of feasible solutions was increased to the maximum value around the 100th iteration. In the later stage of calculation, the number of feasible solutions obtained by the HTS algorithm accounted for about 50% of the total number of members, and keptdecreasing at the later stages of the calculation. By contrast, the number of feasible solutions obtained by the MHTS–TR algorithm accounted for almost 70% of the total number of members, and was relatively stable. This implies that the improvements on the proposed MHTS–TR approach could greatly increase the population density and diversity, and improve the exploration and exploitation capabilities of the population in the feasible region.

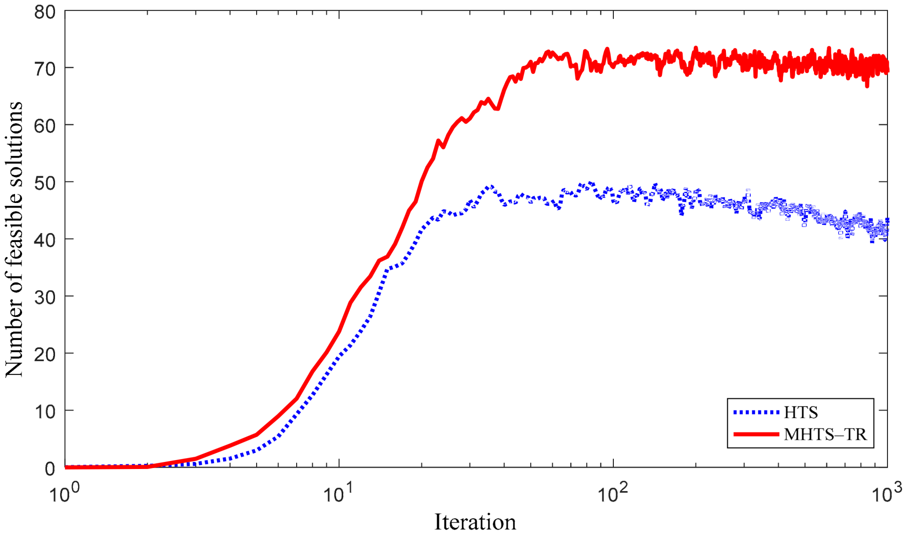

Figure 5. The number of feasible solutions obtained by the original HTS and MHTS–TR algorithms on benchmark function C10.

6. Application to Chemical Constrained Process Optimization

In this section, we further assessed the overall performance of the MHTS–TR approach by applying it to a real-world chemical COP, which was the simplified alkylation process. Moreover, we compared the experimental results obtained by the MHTS–TR approach for this problem with those of other optimization approaches such as the α-based branch and bound method (αBB) [32], the constrained ant colony system (CACS) [33], cultural algorithm with evolutionary programming (CAEP) [34], branch and reduce optimization navigator (BARON) [35] and the basic HTS algorithm. Here, BARON version 21.1.13 with default options was used. The first three methods were tested on this chemical COP previously in the literature; hence, they were employed for comparison with the new variant. To be consistent with the considered competitors, the maximum number of iterations (*maxIter*) was set to 1000, and the population size (*NP*) was set at 50. A detailed description of this chemical process is given in the subsequent sub-section.

6.1. Problem Description

The simplified alkylation process was firstly discussed in detail by Bracken and McCormick [36]. The flowsheet of this process is shown in Figure 6. In this process, an olefin feed of 100% butane, a 100% isobutane make-up stream, and a pure isobutane recycle were infused into a reactor. Then, these components reacted with an acid catalyst, and some spent acid was discharged out of the reactor. Moreover, the product in the reactor was transported into a fractionator, and separated into isobutane and alkylate products.

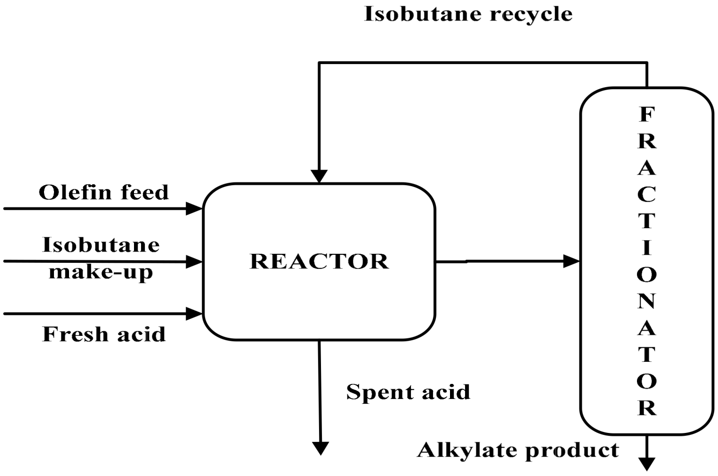

Figure 6. The flowsheet of the simplified alkylation process.

Bracken and McCormick [36] proposed an optimizing model for the alkylation process, and the objective was to maximize the profit. Dembo [37] transformed this process into a model with 7 variables and, subsequently, a slightly modified version of this model was proposed by other researchers in the literature [32]. The variables of this process are listed in Table 6.

Table 6. The variables in the profit-maximization problem of the alkylation process.

Variable	Variable Description	Unit
x_1	Olefin feed rate	Barrel/Day
x_2	Acid addition rate	Kilopound/Day
x_3	Alkylate yield	Barrel/Day
x_4	Acid strength	%
x_5	Motor octane number	-
x_6	External isobutane-to-olefin ratio	%
x_7	F$-$4 performance number	-

The profit-maximization problem for this process takes the form as follows [32]:

$$
\begin{aligned}
&Profit = -\min(1.715\mathrm{x}_1 + 0.035\mathrm{x}_1\mathrm{x}_6 + 4.0565\mathrm{x}_3 + 10.0\mathrm{x}_2 - 0.063\mathrm{x}_3\mathrm{x}_5)\\
&\text{s.t.}\\
&\begin{cases}
g_1 = 0.0059553571x_6^2x_1 + 0.88392857x_3 - 0.1175625x_6x_1 - x_1 \le 0,\\
g_2 = 1.1088x_1 + 0.1303533x_1x_6 - 0.0066033x_1x_6^2 - x_3 \le 0,\\
g_3 = 6.66173269x_6^2 + 172.39878x_5 - 56.596669x_4 - 191.20592x_6 - 10,000 \le 0,\\
g_4 = 1.08702x_6 + 0.32175x_4 - 0.03762x_6^2 - x_5 + 56.85075 \le 0,\\
g_5 = 0.006198x_7x_4x_3 + 2462.3121x_2 - 25.125634x_2x_4 - x_3x_4 \le 0,\\
g_6 = 161.18996x_3x_4 + 5000.0x_2x_4 - 489,510.0x_2 - x_3x_4x_7 \le 0,\\
g_7 = 0.33x_7 - x_5 + 44.333333 \le 0,\\
g_8 = 0.022556x_5 - 0.007595x_7 - 1 \le 0,\\
g_9 = 0.00061x_3 - 0.0005x_1 - 1 \le 0,\\
g_{10} = 0.819672x_1 - x_3 + 0.819672 \le 0,\\
g_{11} = 24,500.0x_2 - 250.0x_2x_4 - x_3x_4 \le 0,\\
g_{12} = 1020.4082x_4x_2 + 1.2244898x_3x_4 - 100,000x_2 \le 0,\\
g_{13} = 6.25x_1x_6 + 6.25x_1 - 7.625x_3 - 100,000 \le 0,\\
g_{14} = 1.22x_3 - x_6x_1 - x_1 + 1 \le 0,\\
1500 \le x_1 \le 2000, 1 \le x_2 \le 120, 3000 \le x_3 \le 3500,\\
85 \le x_4 \le 93, 90 \le x_5 \le 95, 3 \le x_6 \le 12, 145 \le x_7 \le 162.
\end{cases}
\end{aligned}
\tag{15}
$$

6.2. Simulation Results and Discussion

The best values of the profit objective obtained by each approach are displayed in Table 7. The mean execution times of the original HTS and the MHTS–TR approaches on this problem were as follows: the original HTS algorithm spent 12.38 s, and the MHTS–TR approach spent 14.21 s to get these results. It can be observed from Table 7 that the optimization result obtained by the MHTS–TR approach was better than those of the basic HTS, BARON, CACS (δ = 5 × 10^{-6}), and CACS (δ = 0) methods. Moreover, the results obtained by the αBB, CAEP, and CACS (δ = 5 × 10^{-4}) methods were superior to that of the MHTS–TR, but the MHTS–TR method did not violate any constraint. The violations of constraints for the best solutions generated by all algorithms are listed in Table 8. It can be observed from the results table that the αBB, CAEP, CACS (δ = 0), and CACS (δ = 5 × 10^{-4}) approaches violated at least one of the constraints, whereas the CACS (δ = 5 × 10^{-6}), BARON, HTS, and MHTS–TR approaches did not violate any constraints. In comparison with the original HTS, BARON, and CACS (δ = 5 × 10^{-6}) methods, the maximum profit obtained by the MHTS–TR approach was superior. Figure 7 plots the convergence graphs of the optimization results generated by the original HTS and MHTS–TR algorithms. It can be seen from Figure 7 that the proposed MHTS–TR approach showed a superior profit objective compared to the basic HTS method. In the convergence process, due to the appropriate trade-off between exploration and exploitation in our approach being performed by integrating different search mechanisms, the MHTS–TR algorithm converged to a feasible optimum very rapidly, meaning that the overall performance of the MHTS–TR approach was enhanced through the proposed modifications. In summary, the experimental results obtained by the MHTS–TR algorithm on this problem were better than those of the original HTS algorithm and the other competitors. Therefore, we can conclude that the MHTS–TR algorithm is applicable for solving real-world COPs.

Table 7. The comparison results obtained by the αBB, CAEP, CACS, BARON, HTS, and MHTS–TR methods.

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	*Best*
αBB	1698.180	53.660	3031.300	90.110	95.000	10.500	153.530	1772.8
CAEP	1699.8	53.321	3033.1	90.225	95.000	10.485	154.53	1777.1
CACS ($\delta = 0$)	1698.8	54.178	3031.5	90.137	94.992	10.535	153.51	1763.1
CACS ($\delta = 5 \times 10^{-4}$)	1700.4	53.360	3034.7	90.183	94.999	10.322	153.66	1776.6
CACS ($\delta = 5 \times 10^{-6}$)	1700.6	54.346	3033.2	90.183	94.999	10.510	153.53	1763.8
BARON	1698.256	54.274	3031.357	90.190	95.000	10.504	153.535	1766.3
HTS	1701.43	57.81	3031.99	90.23	94.40	10.812	153.72	1592.5
MHTS–TR	1698.11	54.323	3031.3	90.197	95.000	10.497	153.54	1766.4

Table 8. The violations of constraints for the αBB, CAEP, CACS, BARON, HTS, and MHTS–TR methods.

C	αBB	CAEP	CACS ($\delta = 0$)	CACS ($\delta = 5 \times 10^{-4}$)	CACS ($\delta = 5 \times 10^{-6}$)	BARON	HTS	MHTS–TR
g_1	1.650×10^{-2}	-1.1375	-3.266×10^{-1}	-2.4301	-1.9938	0.000	-29.118	-9.3367×10^{-5}
g_2	-60.341	-59.098	-59.965	-57.700	-58.150	-60.324	-60.322	-21.356
g_3	4.7521	-9.854×10^{-1}	5.72×10^{-2}	9.7923	-6.43×10^{-2}	-33.372	-1.1823×10^{-3}	-9.8021×10^{-4}
g_4	-1.8903	-1.8577	-1.8632	-1.9198	-1.8628	-1.863	-1.8633	-1.7981
g_5	-2588.610	-1138.5	-2561.4	-2551.0	-2571.3	-2579.163	-3067.8	$-2579.\ 2$
g_6	1727.870	-2.2415×10^{5}	-4909.4	1357.8	-2154.9	-7.45058×10^{-8}	-29.749	-5.155×10^{-1}
g_7	-1.7670×10^{-3}	3282×10^{-1}	-3.6700×10^{-4}	4.210×10^{-2}	-7.6700×10^{-4}	0.000	-1.0018×10^{-5}	-8.4807×10^{-6}
g_8	-2.320×10^{-2}	-3.080×10^{-2}	-2.330×10^{-2}	-2.430×10^{-2}	-2.330×10^{-2}	-2.30×10^{-2}	-2.4016×10^{-2}	-2.30×10^{-2}
g_9	3.0000×10^{-6}	2.9100×10^{-4}	-1.8500×10^{-4}	9.6700×10^{-4}	-4.8000×10^{-5}	0.000	-1.0440×10^{-7}	-5.5867×10^{-8}
g_{10}	-1638.5	-1639.0	-1638.2	-1640.1	-1638.5	-1638.525	-1636.7	-1638.5
g_{11}	-1.6731×10^{5}	-1.7002×10^{5}	-1.6675×10^{5}	-1.6940×10^{5}	-1.6734×10^{5}	-1.6743×10^{5}	-1.3972×10^{5}	-1.6744×10^{5}
g_{12}	-9.7548×10^{4}	-8.7936×10^{4}	-1.0010×10^{5}	-9.0511×10^{4}	-9.8542×10^{4}	-9.7747×10^{4}	-2.1014×10^{5}	-9.7758×10^{4}
g_{13}	-1057.0	-1113.6	-642.32	-2815.0	-791.24	-1.1282×10^{4}	-2.0265×10^{4}	-1091.2
g_{14}	-1.5830×10^{4}	-1.5821×10^{4}	-1.5896×10^{4}	-1.5549×10^{4}	-1.5872×10^{4}	-1.5837×10^{4}	-1.5824×10^{4}	-1.2962×10^{4}

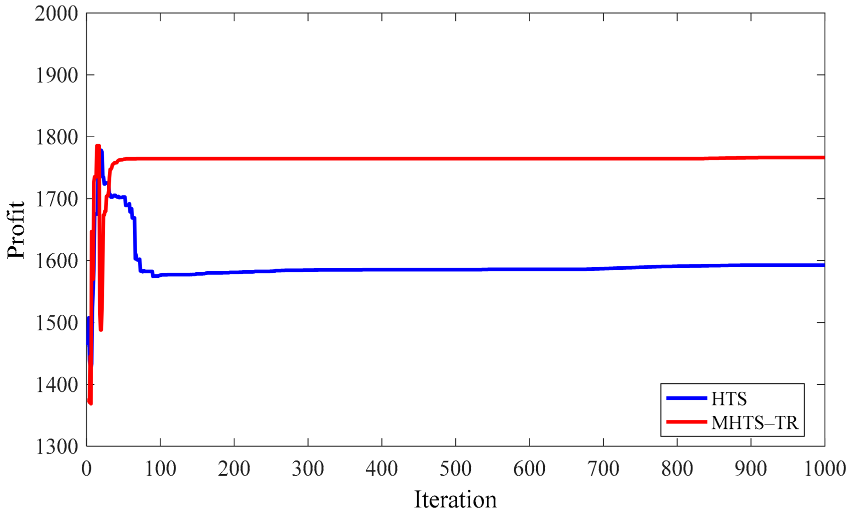

Figure 7. Convergence graph of the original HTS and MHTS–TR algorithms for the simplified alkylation process.

7. Conclusions

Many real-world COPs are defined by complex mathematical equations with different constraints, and simply finding a feasible solution for such problems is not a straightforward task. Thus, to handle COPs efficiently, a novel approach with two search phases called MHTS–TR was proposed in this paper. The feasible search phase (the leader phase) ensured an intensified optimum in a relevant feasible region using the heat transfer search (HTS) algorithm, whereas the infeasible search phase (the follower phase) was used to introduce more diversification into the feasible search phase using the moving mechanism of the tandem running (TR) strategy.

To demonstrate the ability of the proposed MHTS–TR approach on handling different COPs, it was applied to a set of 24 constrained benchmark functions of CEC 2006, which involved different types of functions, such as, non-linear, linear, quadratic, polynomial, and cubic. Moreover, the simulation results were compared with those of the original HTS and other comparative algorithms. The obtained results showed that the proposed MHTS–TR approach markedly outperformed the other comparative algorithms. Finally, to validate the applicability of the MHTS–TR approach in solving real-world COPs, it was applied to handle a chemical COP, which was the simplified alkylation process. Moreover, the simulation results existing in the literature were introduced to be compared with the results obtained by our approach. The MHTS–TR algorithm demonstrated a superior result on the profit value compared to the original HTS method and some other competitors. Moreover, it showed competitive results in satisfying the constraints; therefore, we can conclude that the MHTS–TR approach is applicable for handling real-world COPs. The priority of each MHA is providing good solutions for specific problems, and therefore may not be as effective in providing solutions for some other problems. Thus, the considered application is intended for use on small-scale problems, and may not indicate towards the usefulness of the proposed approach for large-scale problems.

The future paths of research contain the application of the HTS-based optimization method for handling complex chemical engineering problems, such as multi-objective optimization problems involving a larger number of objectives and constraints. Moreover, to further check the robustness and suitability of the proposed method, it can be extended for solving other engineering processes, including single and multi-objective problems.

Author Contributions: Conceptualization, E.A.; methodology, E.A.; software, E.A.; validation, E.A.; formal analysis, E.A.; investigation, E.A.; resources, H.S.; data curation, E.A.; writing—original draft preparation, E.A.; writing—review and editing, E.A.; visualization, H.S.; supervision, H.S.; project administration, H.S.; funding acquisition, H.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by the National Natural Science Foundation of China under Grant 62103149, Grant 62073140 and grant 62073141, in part by National Natural Science Foundation of Shanghai under Grant 19ZR1473200.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. De Castro Rodrigues, M.; Guimarães, S.; de Lima, B.S.L.P. E-BRM: A constraint handling technique to solve optimization problems with evolutionary algorithms. *Appl. Soft Comput.* **2018**, *72*, 14–29. [CrossRef]
2. Garg, H. A hybrid GSA-GA algorithm for constrained optimization problems. *Inf. Sci.* **2019**, *478*, 499–523. [CrossRef]
3. Bajaj, I.; Iyer, S.S.; Hasan, M.F. A trust region-based two-phase algorithm for constrained black-box and grey-box optimization with infeasible initial point. *Comput. Chem. Eng.* **2018**, *116*, 306–321. [CrossRef]
4. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [CrossRef]
5. Coello, C.A.C. Constraint-Handling Techniques Used with Evolutionary Algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, Lille, France, 10–14 July 2021; pp. 692–714.
6. Jordehi, A.R. A review on constraint handling strategies in particle swarm optimisation. *Neural Comput. Appl.* **2015**, *26*, 1265–1275. [CrossRef]
7. Jan, M.A.; Sagheer, M.; Khan, H.U.; Uddin, M.I.; Khanum, R.A.; Mahmoud, M.; Mast, N. Hybrid stochastic ranking for constrained optimization. *IEEE Access* **2020**, *8*, 227270–227287. [CrossRef]

8. Kaur, S.; Awasthi, L.K.; Sangal, A. Hmoshssa: A hybrid meta-heuristic approach for solving constrained optimization problems. *Eng. Comput.* **2021**, *37*, 3167–3203. [CrossRef]
9. Costa, M.F.P.; Francisco, R.B.; Rocha, A.M.A.; Fernandes, E.M. Theoretical and practical convergence of a self-adaptive penalty algorithm for constrained global optimization. *J. Optim. Theory Appl.* **2017**, *174*, 875–893. [CrossRef]
10. Wang, B.-C.; Li, H.-X.; Li, J.-P.; Wang, Y. Composite differential evolution for constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 1482–1495. [CrossRef]
11. Babalik, A.; Cinar, A.C.; Kiran, M.S. A modification of tree-seed algorithm using deb's rules for constrained optimization. *Appl. Soft Comput.* **2018**, *63*, 289–305. [CrossRef]
12. Mehlitz, P. A comparison of solution approaches for the numerical treatment of or-constrained optimization problems. *Comput. Optim. Appl.* **2020**, *76*, 233–275. [CrossRef]
13. Ang, K.M.; Lim, W.H.; Isa, N.A.M.; Tiang, S.S.; Wong, C.H. A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems. *Expert Syst. Appl.* **2020**, *140*, 112882. [CrossRef]
14. Chen, Y.; Hao, J.-K.; Glover, F. A hybrid metaheuristic approach for the capacitated arc routing problem. *Eur. J. Oper. Res.* **2016**, *253*, 25–39. [CrossRef]
15. Paraskevopoulos, D.C.; Laporte, G.; Repoussis, P.P.; Tarantilis, C.D. Resource constrained routing and scheduling: Review and research prospects. *Eur. J. Oper. Res.* **2017**, *263*, 737–754. [CrossRef]
16. Errico, F.; Desaulniers, G.; Gendreau, M.; Rei, W.; Rousseau, L.-M. A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times. *Eur. J. Oper. Res.* **2016**, *249*, 55–66. [CrossRef]
17. Georgioudakis, M.; Plevris, V. A comparative study of differential evolution variants in constrained structural optimization. *Front. Built Environ.* **2020**, *6*, 102. [CrossRef]
18. Patel, V.K.; Savsani, V.J. Heat transfer search (HTS): A novel optimization algorithm. *Inf. Sci.* **2015**, *324*, 217–246. [CrossRef]
19. Alnahari, E.; Shi, H.; Alkebsi, K. Quadratic interpolation based simultaneous heat transfer search algorithm and its application to chemical dynamic system optimization. *Processes* **2020**, *8*, 478. [CrossRef]
20. Tejani, G.G.; Savsani, V.J.; Patel, V.K.; Mirjalili, S. An improved heat transfer search algorithm for unconstrained optimization problems. *J. Comput. Des. Eng.* **2019**, *6*, 13–32. [CrossRef]
21. Tejani, G.G.; Kumar, S.; Gandomi, A.H. Multi-objective heat transfer search algorithm for truss optimization. *Eng. Comput.* **2019**, *37*, 641–642. [CrossRef]
22. Degertekin, S.; Lamberti, L.; Hayalioglu, M. Heat transfer search algorithm for sizing optimization of truss structures. *Lat. Am. J. Solids Struct.* **2017**, *14*, 373–397. [CrossRef]
23. Tejani, G.; Savsani, V.; Patel, V. Modified sub-population-based heat transfer search algorithm for structural optimization. *Int. J. Appl. Metaheuristic Comput. (IJAMC)* **2017**, *8*, 1–23. [CrossRef]
24. Tawhid, M.A.; Savsani, V. ∈-constraint heat transfer search (∈-HTS) algorithm for solving multi-objective engineering design problems. *J. Comput. Des. Eng.* **2018**, *5*, 104–119. [CrossRef]
25. Maharana, D.; Kotecha, P. Simultaneous heat transfer search for computationally expensive numerical optimization. In Proceedings of the 2016 IEEE Congress on evolutionary computation (CEC), Vancouver, BC, Canada, 24–29 July 2016; pp. 2982–2988.
26. Maharana, D.; Kotecha, P. In Simultaneous heat transfer search for single objective real-parameter numerical optimization problem. In Proceedings of the 2016 IEEE Region 10 Conference (TENCON), Marina Bay Sands, Singapore, 22–26 November 2016; pp. 2138–2141.
27. Lee, J. Leader-member exchange, the "pelz effect," and cooperative communication between group members. *Manag. Commun. Q.* **1997**, *11*, 266–287. [CrossRef]
28. Franklin, E.L.; Richardson, T.O.; Sendova-Franks, A.B.; Robinson, E.J.; Franks, N.R. Blinkered teaching: Tandem running by visually impaired ants. *Behav. Ecol. Sociobiol.* **2011**, *65*, 569–579. [CrossRef]
29. Karaboga, D.; Akay, B. A modified artificial bee colony (ABC) algorithm for constrained optimization problems. *Appl. Soft Comput.* **2011**, *11*, 3021–3031. [CrossRef]
30. Karaboga, D.; Basturk, B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In Proceedings of the International Fuzzy Systems Association World Congress, Cancun, Mexico, 18–21 June 2007; pp. 789–798.
31. Liang, J.; Runarsson, T.P.; Mezura-Montes, E.; Clerc, M.; Suganthan, P.N.; Coello, C.C.; Deb, K. Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization. *J. Appl. Mech.* **2006**, *41*, 8–31.
32. Adjiman, C.S.; Dallwig, S.; Floudas, C.A.; Neumaier, A. A global optimization method, αbb, for general twice-differentiable constrained NLPs—I. Theoretical advances. *Comput. Chem. Eng.* **1998**, *22*, 1137–1158. [CrossRef]
33. He, Y.; Chen, D. Constrained ant colony system and its application in process optimization of butene alkylation. *J. Chem. Ind. Eng.–China* **2005**, *56*, 1708.
34. Huang, H.-y.; Gu, X.-s.; Liu, M.-d. Research on cultural algorithm for solving nonlinear constrained optimization. *Acta Autom. Sin.* **2007**, *33*, 1115.
35. Akteke Ozturk, B.; Koksal, G.; Weber, G.W. Optimization of desirability functions as a DNLP model by GAMS/BARON. In Proceedings of the AIP Conference Proceedings, Chonburi, Thailand, 18–20 July 2018; pp. 305–310.
36. Bracken, J.; McCormick, G.P. *Selected Applications of Nonlinear Programming*; Research Analysis Corporation: McLean, VA, USA, 1968.
37. Dembo, R.S. A set of geometric programming test problems and their solutions. *Math. Program.* **1976**, *10*, 192–213. [CrossRef]