SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2021, Vol. 64 150101:1–150101:12
https://doi.org/10.1007/s11432-020-3115-5

Special Focus on Constraints and Optimization in Artificial Intelligence

Focal distance tabu search

Fred GLOVER[1*] & Zhipeng LÜ[2*]

[1]*Meta-Analytics, Inc., Boulder,* CO 80309*, USA;*
[2]*School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan* 430074*, China*

Received 10 May 2020/Accepted 4 August 2020/Published online 12 April 2021

Abstract Focal distance tabu search modifies a standard tabu search algorithm for binary optimization by augmenting a periodic diversification step that drives the search away from a current best (or elite) solution until the objective function deteriorates beyond a specified threshold or until attaining a lower bound on the distance from the originating solution. The new augmented algorithm combines the threshold and lower bound approaches by introducing an initial focal distance for the lower bound which is updated when the diversification step is completed. However, rather than terminating the diversification step at the customary completion point, focal distance tabu search (TS) retains the focal distance bound through additional search phases designed to improve the objective function, drawing on a strategy proposed with strategic oscillation. The algorithm realizes this strategy by partitioning the variables into two sets which are managed together with an abbreviated tabu search process. An advanced version of the approach drives the search away from a collection of solutions rather than a single originating solution, introducing the concept of a signature solution to guide the search. The method can be employed to augment a variety of other metaheuristic algorithms such as those using threshold procedures, late acceptance hill climbing, iterated local search, breakout local search, GRASP, and path relinking.

Keywords tabu search, diversification, strategic oscillation, adaptive partitioning, metaheuristics

Citation Glover F, Lü Z P. Focal distance tabu search. Sci China Inf Sci, 2021, 64(5): 150101, https://doi.org/10.1007/s11432-020-3115-5

1 Background

Focal distance tabu search introduces a new diversification component that augments the customary tabu search (TS) diversification approach of driving the search away from a current solution when progress slows. A key element of focal distance TS is to provide a modified form of search that maintains the search for a chosen period at a specified distance that operates as lower limit of separation from its origin.

The algorithm builds on classical TS threshold and strategic oscillation approaches by driving the solution progressively farther away from a selected solution when successive diversification efforts are unproductive. Focal distance TS goes farther in two ways, first by maintaining the search for a period in a new region that is removed from a single origin, and second by a more advanced procedure that maintains the search in a region that is removed from a collection of origins. By this latter procedure, the new search region is prevented from cycling back through previous regions as it departs from previous origins and can search new regions that are diverse relative to each other.

The present paper focuses on binary optimization, but our approach can be modified to apply to optimization in other settings. The optimization problem of interest may be written as

$$\text{(P)} \quad \min x_o = f(x) \quad \text{s.t. } x \in X \text{ and } x \text{ binary}, \tag{1}$$

where the vector $x = (x_1, \ldots, x_n)$ is alternately expressed as $x = (x_j : j \in N)$ for $N = \{1, \ldots, n\}$.

We assume that the structure of X permits a neighborhood search for transitioning from one binary solution $x \in X$ to another by rules that identify neighbor solutions. We further suppose these transitions

* Corresponding author (email: fredwglover@yahoo.com, zhipeng.lv@hust.edu.cn)

result by "flipping" variables using 1-flips or 2-flips, which respectively consist of changing the value of one or two binary variables from 0 to 1 or from 1 to 0. Higher order flips are possible but not essential.

An instance of (P) that has attracted considerable attention in recent years is the quadratic unconstrained binary optimization problem:

$$\text{(QUBO)} \quad \min x_o = xQx \quad \text{s.t. } x \text{ binary}, \tag{2}$$

where Q is an $n \times n$ matrix. Examples of the wide range of applications embraced by the QUBO model are found in the surveys by Lucas [1], Kochenberger et al. [2] and Glover et al. [3]. Our methodology in this paper is particularly relevant to QUBO problems because of the freedom of choosing variables to flip for a model whose only constraint is to require that variables are binary. However, there are QUBO variants that also involve additional constraining relationships [2, 4], and our approach can be applied to them as well.

2 Previous related methods

Classical tabu threshold approaches for diversification typically employ the simple idea of starting from a high quality (usually a current best) solution for (P) and flipping variables by a chosen strategy until yielding a solution that satisfies a threshold on the objective function x_o.

Control of the solutions by reference to objective function values was proposed in [5] through the notion of "adaptive thresholding" as an alternative to simulated annealing, extending strategic oscillation (oscillating assignment) from [6] which represents a fundamental component of tabu search. The application of thresholding to the objective function, without strategic oscillation, was introduced in [7] and later, by including a rudimentary form of oscillation in [8]. Related ideas were developed in the great deluge approach of Dueck [9]. A more comprehensive use of strategic oscillation for controlling the objective function was introduced in the tabu thresholding approach of Glover [10] and the critical event memory approach of Glover and Kochenberger [11].

Two early variations of tabu search that employ strategies for driving the search away from a current solution without using objective function thresholds were proposed in [12, 13]. The former approach flipped variables randomly for a randomly chosen (progressively enlarging) number of iterations utilizing a frequency memory and hash function ideas from Woodruff and Zemel [14], while the latter approach instead achieved diversification by selecting best non-tabu moves at each iteration, imposing an arbitrarily large tabu tenure until no more moves were possible, whereon all tabu restrictions were removed to descend to a local optimum before repeating.

We propose a different type of tabu diversification based on selecting values for a focal distance as a lower bound for separation but does not terminate the diversification process once the desired separation is attained. Instead focal distance tabu search introduces a variable partitioning strategy that divides the variables into two sets to guide the search by controlling the transitions from one set to the other. Strategic oscillation is invoked by moving away from a solution to varying levels and launching an improving search from each level, accompanied by the focal distance partitioning approach. A basic version of the method operates by expressing the focal distance in terms of separation from a single solution x^*, while a more advanced version adds a further dimension by driving the solution a chosen distance from a collection X^* of solutions, and then maintains this distance through the diversification phases that follow. This ability to simultaneously drive the search away from multiple regions avoids the risk that moving away from a given region will simply move back to a region previously visited. The variable partitioning strategy makes the method easy to execute with different options for selecting the focal distance values. We first sketch the method in overview, and then give a more detailed description.

3 Overview: moving away from a single solution x^*

Let $x^{\#}$ denote a solution produced by starting from a chosen solution x^* and flipping variables for some chosen number of steps. The classical threshold condition for terminating the flipping consists of stopping when the objective value $x_o^{\#}$ for $x^{\#}$ satisfies $x_o^{\#} \geqslant x_o\text{Thresh}$. For diversification purposes the goal is to select the threshold $x_o\text{Thresh}$ large enough so that $x_o^{\#} \geqslant x_o\text{Thresh}$ assures the current solution $x^{\#}$ is "far enough" away from x^* when stopping so that the search will enter a productive new region. But it is also important to keep from selecting x_oThresh unduly large, which is reinforced by the fact that a smaller x_oThresh value can enable re-optimization starting from $x^\#$ to be performed more quickly than re-starting from a solution that is farther away. Consequently, the approach is often employed by first choosing x_oThresh to be a modest value and then increasing the value if the solution process following diversification brings the current solution back to the vicinity of the original x^*.

Although the basic approach has been used in a variety of tabu search papers, little experimentation has been devoted to the question of how to choose x_oThresh. In most cases the variables to be flipped are chosen randomly. A contrasting approach is used an early tabu search paper [15] by employing strategic oscillation to flip variables non-randomly, increasing and decreasing x_oThresh in waves. The method seeks to worsen (increase) x_o by the least amount or to improve (decrease) x_o by the greatest amount at each step, subject to tabu conditions to assure the solutions generated will not duplicate previous ones. However, this approach likewise has not been extensively examined.

The basic structure of focal distance tabu search, restricting attention to driving the current solution away from a single solution x^*, may be sketched as follows.

The method begins by applying tabu search for a chosen number of iterations to obtain the first solution x^*.

A value x_oThresh is then chosen as in the classical approach. A preliminary focal distance FocalD is also selected to assure that $x^\#$ lies sufficiently far from x^* by requiring $||x^\# - x^*|| \geqslant$ FocalD, where $||x^\# - x^*||$ is the Hamming distance between $x^\#$ and x^* (or equivalently, the number of flips required to transform x^* to $x^\#$). Relative to a current $x^\#$, we write the value $||x^\# - x^*||$ as the current distance CurrentD. This is useful for generalizations discussed later.

Phase 0. Generate $x^\#$ to satisfy the two conditions $x_o^\# \geqslant x_o$Thresh and CurrentD $\geqslant$ FocalD, for CurrentD $= ||x^\# - x^*||$, by a diversification step that flips successive variables at random or probabilistically starting from x^* until the indicated conditions are met. The final value of CurrentD then gives the updated value for FocalD which may be larger than the original value if the x_oThresh threshold condition drives x_o farther from x_o^* than the condition CurrentD $\geqslant$ FocalD based on the original FocalD value.

Remark 1. A probabilistic choice of variables to flip can be based on keeping track of the number r_j of times x_j receives the value 1 in r local optima encountered while obtaining x^*. Then, starting from x^*, the probability assigned to flipping x_j^* can be determined by the weight r_j if $x_j^* = 0$ and the weight $(r - r_j)$ if $x_j^* = 1$, adjusted so that each step flips a variable not previously flipped during the diversification step. For greater diversification, the weights for flipping x_j^* can be reversed by using the weight r_j if $x_j^* = 1$ and the weight $(r - r_j)$ if $x_j^* = 0$. The probabilities can also be determined by evaluating the change in x_o produced by each successive flip during the diversification, although this is slower. Satisfying the conditions $x_o^\# \geqslant x_o$Thresh and $||x^\# - x^*|| \geqslant$ FocalD may be more important than using probabilistic weights, and hence in some cases random flipping may be sufficient.

In contrast to the classical diversification approach, focal distance tabu search does not end the diversification process after applying (some version of) Phase 0, but instead employs additional phases to progressively improve the solution $x^\#$ subject to satisfying the separation requirement CurrentD $\geqslant$ FocalD. We briefly sketch an outline of these additional phases and then subsequently describe them in detail.

Phase 1. Introduce a variable partitioning method to give a variant of tabu search designed to quickly find a local optimum x^L such that CurrentD $\geqslant$ FocalD.

Remark 2. No consideration is given here to satisfying $x_o^L \geqslant x_o$Thresh. Instead, the search is just looking for a good solution x^L satisfying CurrentD $\geqslant$ FocalD, and the smaller the value of x_o^L the better. For this reason, a possible variant of Phase 0 is to eliminate the initial requirement $x_o^\# \geqslant x_o$Thresh and simply seek to satisfy CurrentD $\geqslant$ FocalD, before proceeding to improve $x^\#$ to obtain a solution x^L.

Remark 3. When $x_o^\# \geqslant x_o$Thresh is eliminated in Phase 0, the value of FocalD can be determined by running preliminary trials using $x_o^\# \geqslant x_o$Thresh as the stopping criterion, and then setting FocalD to the Mean, Max or 0.5(Mean + Max) of the value CurrentD $= ||x^\# - x^*||$ obtained from these trials. Thereafter, the criterion CurrentD $\geqslant$ FocalD can replace $x_o^\# \geqslant x_o$Thresh. Alternatively, experiments can also be performed to select FocalD as a specified fraction of n, as described subsequently. However, for some kinds of problems a Hamming distance measure for CurrentD is inappropriate, and in these cases, it can be useful to retain $x_o^\# \geqslant x_o$Thresh in Phase 0.

Phase 2. Execute the variable partitioning approach of Phase 1 in conjunction with tabu search for a limited number of iterations, LimIter, and use a special initialization of tabu tenures, to improve the solution x^L from Phase 1 further, again subject to CurrentD $\geqslant$ FocalD.

Phase 3. Take the best solution from Phase 2 as the starting solution for a customary tabu search approach, beginning with all tabu tenures 0 (no variables are tabu to flip) and disregarding the requirement CurrentD $\geqslant$ FocalD.

The parallel processing step that carries out multiple instances of the sequence Phase 0 $\rightarrow$ Phase 1 $\rightarrow$ Phase 2 $\rightarrow$ Phase 3 simultaneously is called a Round. A new solution x^* is identified as the best solution obtained from the parallel runs of the Round, and in coordination with other updates, including FocalD, a new Round is launched to repeat the foregoing process.

When x^* is not improved at the conclusion of a Round (or after a specified number of Rounds), then x_oThresh and FocalD are increased in a focal adjustment step to drive the solution farther from x^* in Phase 0, and the series of preceding Rounds are run again. Overall termination occurs when the focal adjustment step determines that no gains are to be expected from continuing to increase x_oThresh and FocalD.

Details of the algorithm are as follows.

4 Focal distance tabu search for a single solution x^*

Apply a tabu search algorithm to problem (P) for a chosen number of iterations and let x^* denote the best solution found. Then execute the following.

Choose an objective function threshold $x_o\text{Thresh} > x_o^*$ and a preliminary value for the distance FocalD. The value FocalD will normally be a fraction of n lying in an interval such as $[0.1n, 0.25n]$, though for greater diversification FocalD may be chosen larger. The objective function threshold x_oThresh may be chosen as a fraction of x_o^*; for example, assuming $x_o^* < 0$, $x_o\text{Thresh} = \text{fraction} \times x_o^*$ where fraction $= 0.8, 0.75$, etc.

Remark 4. In contrast to the approach of setting $x_o\text{Thresh} = \text{fraction} \times x_o^*$ for $x_o = f(x)$, it may be appropriate to define the objective as $x_o = c_o + f(x)$ where c_o is chosen as follows. Let fAvg denote the average of the $f(x)$ values over a collection of randomly generated solutions, or the average of the $f(x)$ values over locally optimal solutions starting from a collection of randomly generated solutions. Then c_o is chosen so that $c_o + f\text{Avg} = 0.5(c_o + f(x^*))$ giving $c_o = f(x^*) - 2f\text{Avg}$, and hence $x_o^* = c_o + f(x^*) = 2(f(x^*) - f\text{Avg})$. For example, if fraction $= 0.75$, then $x_o\text{Thresh} = 1.5(f(x^*) - f\text{Avg})$ and in general $x_o^{\#} \geqslant x_o\text{Thresh}$ gives $f(x^{\#}) \geqslant x_o\text{Thresh} - c_o$.

Then execute the following phases in parallel for a chosen number of threads.

Phase 0. The goal of Phase 0 is to perform a diversification step to drive the search from x^* to a new solution $x^{\#}$ such that

$$\text{CurrentD} \geqslant \text{FocalD and } x_o^{\#} \geqslant x_o\text{Thresh} \tag{A}$$

followed by updating the focal distance relating $x^{\#}$ and x^* by setting FocalD = CurrentD at the conclusion of Phase 0.

Remark 5. The diversification may be performed as noted earlier by executing a collection of 1-flips starting with x^* and continuing until Eq. (A) is satisfied, where no 1-flip is permitted to be reversed. The 1-flips may be chosen randomly or probabiltistically as described in Remark 1.

Variable partitioning. To execute Phase 0 in a form that sets the stage for subsequent phases of the algorithm, the variables are partitioned by reference to their index set $N = \{1, \dots, n\}$ to create the two subsets $N(*) = \{j \in N : x_j^{\#} = x_j^*\}$ and $N(\sim *) = \{j \in N : x_j^{\#} \neq x_j^*\} = \{j \in N : x_j^{\#} = 1 - x_j^*\}$. The partitioning can be established in the execution of Phase 0 by expressing this phase in the following way.

Initialization

Given the current values of FocalD and x_oThresh, set $x^{\#} = x^*$ and CurrentD = 0, and set $N(*) = N$ and $N(\sim *) = \emptyset$.

Main routine

While CurrentD < FocalD and $x_o^{\#} < x_o$Thresh;

Choose $j \in N(*)$ and flip $x_j^{\#}$ (which equals x_j^* by the definition of $N(*)$);

$x_j^{\#} := 1 - x_j^{\#}$;

$N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$;

CurrentD := CurrentD + 1;

Endwhile

Remark 6. At the conclusion of the main routine, CurrentD $\geqslant$ FocalD, where we use the Hamming distance measure for CurrentD that increases by 1 at each 1-flip of an element $j \in N(*)$. Consequently, for the final solution x^L we have CurrentD $= ||x^L - x^*||$. Later we consider a generalization using a distance measure where CurrentD can increase by a value different than 1.

To launch subsequent phases, update FocalD = CurrentD, $x^L = x\text{Best} = x^{\#}$, hence, implicitly, $x_o^L = x_o\text{Best} = x_o^{\#}$. Note the partitioning of N implies $|N(\sim *)| =$ FocalD at the end of Phase 0. In the first execution of Phase 0 by the focal distance TS method, FocalD will typically be less than $0.5n$ and hence the set $N(\sim *)$ will normally be somewhat smaller than $N(*)$.

Phase 1 (Quick descent to a local optimum satisfying CurrentD $\geqslant$ FocalD).

Initialization From Phase 0, adopt the final partitioning of N into $N(*)$ and $N(\sim *)$, as well as the final assignment FocalD = CurrentD, and let $x = x^{\#}$ and Stop = False.

Main routine

While Stop = False

1. Choose a highest evaluation 1-flip of a variable $x_j, j \in N$.
   If this flip does not improve (decrease) x_o, then set Stop = True, set $x^L = x$ (and $x_o^L = x_o$) and proceed to Phase 2 with a local D-optimum x^L. Otherwise,
2. If $j \in N(*)$ or if CurrentD > FocalD, execute the flip to produce a new solution, again denoted by x.
   (a) If $j \in N(*)$, let CurrentD := CurrentD + 1 and move j from $N(*)$ to $N(\sim *)$, giving $N(*) := N(*) \backslash \{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$.
   (b) Else if $j \in N(\sim *)$, let CurrentD := CurrentD − 1 and move j from $N(\sim *)$ to $N(*)$, giving $N(\sim *) := N(\sim *) \backslash \{j\}$ and $N(*) := N(*) \cup \{j\}$.
3. Else if CurrentD = FocalD($j \in N(\sim *)$), choose a highest evaluation 1-flip of a variable $x_j, j \in N(*)$.
   (a) If this 1-flip improves x_o then execute the flip to produce a new solution, again denoted by x, and execute 2(a).
   (b) Else (the 1-flip is non-improving) set x_oSave $= x_o$, and execute the 1-flip, denoting the new solution by x, and execute 2(a), resulting in CurrentD > FocalD.
   Choose a highest evaluation 1-flip of a variable x_j, $j \in N$ as in Step 1.
      i. If this flip does not yield a value for $x_o < x_o$Save, recover the previous x by reversing the 1-flip that produced the new x in 3(b), reverse the update of $N(*)$ and $N(\sim *)$ performed in 2(a) (under 3(b)) and set $x^L = x$, $x_o^L = x_o = x_o$Save and Stop = True. Then proceed to Phase 2 with a local D-optimum x^L.
      ii. Else (the flip yields an improved value for $x_o < x_o$Save): Continue to the next iteration (to repeat 1).

Endwhile

In anticipation of Phase 2, let xBest $= x^L$ (hence x_oBest $= x_o^L$). Note that xBest differs from x^* as a result of satisfying $||x^L - x^*|| \geqslant$ FocalD.

Remark 7. Phase 1 above can be simplified by terminating upon reaching 3(b) without executing this step, at the risk of not reaching a local optimum as good as might otherwise be obtained. As indicated subsequently, the inclusion of 2-flips can also avoid executing 3(b).

Phase 2 (Beginning with a local D-optimum).

Initialization. Let $x = x$Best and $x_o = x_o$Best. Also, for $N(*)$ and $N(\sim *)$ inherited from Phase 1, assign a 0 initial tabu tenure to all x_j for $j \in N(*)$ (hence these variables begin non-tabu), and assign a tabu tenure SmallTenure to all x_j for $j \in N(\sim *)$. For example, SmallTenure can be chosen from the interval $[0, 0.5\text{CurrentD}]$. If SmallTenure = 0, then all variables start non-tabu, and Phase 2 is skipped,

passing directly to Phase 3.
Choose an iteration limit LimIter > SmallTenure. Subsequently, as variables are flipped, they are assigned a tabu tenure by customary tabu search rules.
Main routine
Unlike Phase 1, Phase 2 does not terminate upon reaching a local optimum, but rather depends on LimIter. Nevertheless, Phase 2 is simpler than Phase 1 because tabu restrictions avoid the complications posed by step 3(b). The organization of Phase 2 continues to assure $||x - x^*|| \geqslant$ FocalD.
For Iter = 1 to LimIter

1. Choose a highest evaluation non-tabu 1-flip of a variable $x_j, j \in N$.
2. Execute the flip to produce a new solution, again denoted by x.
   (a) If $j \in N(*)$, let CurrentD := CurrentD + 1 and move j from $N(*)$ to $N(\sim *)$, giving $N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$.
   (b) Else if $j \in N(\sim *)$, let CurrentD := CurrentD − 1 and move j from $N(\sim *)$ to $N(*)$, giving $N(\sim *) := N(\sim *)\backslash\{j\}$ and $N(*) := N(*) \cup \{j\}$.
3. If CurrentD < FocalD, choose a highest evaluation non-tabu 1-flip of a variable $x_j, j \in N(*)$, set CurrentD := CurrentD + 1 and move j from $N(*)$ to $N(\sim *)$ giving $N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$. (Now CurrentD $\geqslant$ FocalD.)
4. If $x_o < x_o$Best, then update x_oBest $= x_o$ and xBest $= x$.

Endfor

Remark 8. Both Phases 1 and 2 can be improved by including 2-flip moves. For example, in Step 3 of Phase 1 where if CurrentD = FocalD and $j \in N(\sim *)$ in Phase 1, the method can check whether a 2-flip move, with at least one of the flips coming from $N(*)$, can improve x_o and terminate if no improvement results. This replacement of Step 3 simplifies Phase 1.

Remark 9. As noted in the Initialization of Phase 2, the algorithm can jump directly from Phase 1 to Phase 3 without executing Phase 2 by setting SmallTenure = 0. Another option is to skip Phase 1 and go directly from Phase 0 to Phase 2. This can be done by performing the Initialization for Phase 1 after Phase 0 and setting $x^L = x$. In this case, a reduced tabu search phase that maintains CurrentD $\geqslant$ FocalD can replace the phase of first proceeding to a local optimum x^L as done in Phase 1. The sequence from Phase 0 to Phase 2, potentially skipping over Phase 1 or Phase 2, can be executed several times in succession for increasing values of x_oThresh and FocalD, to identify values of these parameters that lead to interesting new solution outcomes before continuing to Phase 3. Such an option may be particularly relevant for the version of focal distance TS applied to a set X^*, described below.

Phase 3 (Beginning from a solution xBest from Phase 1 or Phase 2). Apply a customary TS approach starting from $x = x$Best. Begin with all tabu tenures 0 (no variables are tabu to flip) and disregard the requirement CurrentD $\geqslant$ FocalD.

As noted earlier, parallel processing is used to execute a series of Rounds that carry out multiple instances of the sequence Phase 0 → Phase 1 → Phase 2 → Phase 3 simultaneously. At the conclusion of each Round, the best solution x^* from the resulting collection of parallel solution efforts is identified to repeat the foregoing process, while a focal adjustment step increases the values of x_oThresh and FocalD if x^* is not improved by the latest Round (or after a specified number of Rounds).

The following summary of the method puts all these considerations together. We refer to the tabu search approach that launches the method and that is used in Phase 3 as the basic algorithm, and refer to the form of this method used in Phase 2 as the abbreviated basic algorithm. The algorithm of Phase 1 that uses the variable partitioning with local descent will be called the constrained local improvement algorithm. The basic algorithm uses an initial stopping criterion that limits the number of iterations it executes when launching the method, and also uses a Phase 3 stopping criterion that limits the number of iterations it executes in Phase 3. The complete method has an overall termination criterion, based on the number of iterations performed that fail to improve the best solution x^* and a lower limit on the value of x_oThresh and an upper limit on the value of FocalD.

Summary: focal distance tabu search algorithm for a single solution x^*.

Initial step: Apply the basic algorithm until satisfying the initial stopping criterion. Identify the initial best solution x^* and select beginning x_oThresh and FocalD values for Phase 0.
While the overall termination criterion is not satisfied.
  Set Improve = True;
  While Improve = True (Round Loop);
    Carry out a Round consisting of parallel threads initiated by the current x^* and the current x_oThresh and FocalD values as follows:
      Execute the diversification process of Phase 0 and update FocalD;
      Execute the constrained local improvement algorithm of Phase 1;
      Execute the abbreviated basic algorithm of Phase 2;
      Execute the basic algorithm in Phase 3 until satisfying the Phase 3 stopping criterion.
    Identify the best solution x^* obtained over all the threads and the value of FocalD that produced this x^*. If x^* is not improved compared to the x^* that initiated the current Round, then set Improve = False and terminate the Round Loop. Otherwise continue.
  Endwhile
  Focal adjustment step
  If the overall stopping criterion is not satisfied, increase x_oThresh and the value of FocalD inherited from the Round Loop and continue by beginning from the best solution x^* previously found.
Endwhile

The basic algorithm in the initial step can also be implemented using multiple threads, in which case the initial x^* will be the best solution over all threads. Now we describe the more advanced version of focal distance TS that seeks new solutions that lie a minimum distance FocalD from more than one solution x^*. We identify additional options for the single x^* solution case in discussing the analog of the preceding algorithm applied to a set of solutions.

5 Focal distance tabu search for a set X^* of solutions

Focal distance tabu search applied to a set X^* of solutions involves several additional concepts that enable the algorithm to be expressed in a form that resembles the basic focal distance TS applied to a single solution x^*.

The signature vector x^S. In place of the single solution x^* used in the simpler form of focal distance tabu search, we begin Phase 0 by creating a signature vector x^S for the solutions $x \in X^*$, by setting $x_j^S = v$, for $v = 0$ or 1, if the majority of solutions $x \in X^*$ yields $x_j = v$.

To do this, denote the number of solutions in X^* by $m = |X^*|$, and consider the subset of solutions in X^* given by $X^*(j:v) = \{x \in X^* : x_j = v\}$ and represent its cardinality by $m(j:v) = |X^*(j:v)|$. It follows that $m(j:0)+m(j:1) = m$. Let $\text{Max}(j) = \text{Max}(m(j:0), m(j:1))$ and $\text{Min}(j) = \text{Min}(m(j:0), m(j:1))$, which also yields $\text{Max}(j) + \text{Min}(j) = m$. Finally, the signature vector x^S may be defined by

$$x_j^S = 1 \text{ if } m(j:1) > m(j:0), \tag{3}$$

$$x_j^S = 0 \text{ if } m(j:0) > m(j:1). \tag{4}$$

For the case $m(j:1) = m(j:0)$, if there exists any x_j that satisfies this case, we choose x_j^S randomly to be 0 or 1, subject to requiring that half (rounding down if necessary) of the j yielding $m(j:1) = m(j:0)$ receive the value 0 and the remainder receive the value 1. Thus, in all cases, we have $x_j^S = v$ for $\text{Max}(j) = m(j:v)$.

The result of flipping a variable x_j in the signature vector x_j^S from x_j^S to $1 - x_j^S$ will cause the new vector to "move away from" Max(j) different vectors in X^* and "move toward" Min(j) different vectors in X^*. Hence the net diversification effect relative to X^* achieved by this flip will be given by the value d_j identified as

$$d_j = \text{Max}(j) - \text{Min}(j). \tag{5}$$

In the simpler situation where we consider a single solution x^*, i.e., $X^* = \{x^*\}$, it is evident that x^* is also the signature vector x_j^S of X^*, and $\text{Max}(j) - \text{Min}(j) = 0$, hence $d_j = m(= 1)$. In general, for a set X^* containing m solutions, the equality $d_j = m$ holds for each x_j that receives the same value in all $x \in X^*$. The subset $N(j:m)$ of N that contains all variables x_j such that $d_j = m$ (hence for which

x_j receives the same value v for all $x \in X^*$) may be viewed as defining the "intersection" of the vectors $x \in X^*$, where the vectors are treated as sets and the intersection of two sets is defined to be the indexes of all variables that receive the same values in these sets.

The condition $d_j = m$, which means that flipping x_j^S to $1 - x_j^S$ moves away from all solutions in X^*, may be interpreted as having the effect of moving a unit distance away from the full set X^*. In general, we define the distance that flipping x_j^S to $1 - x_j^S$ moves away from X^* to be

$$D_j = d_j/m. \tag{6}$$

Recalling that d_j measures the difference between the number of solutions in X^* that a flip "moves away from" minus the number of solutions in X^* that a flip "moves toward", we see that

$$\text{For Max}(j) = m : d_j = m \text{ (and } D_j = m/m = 1), \tag{7}$$

$$\text{For Max}(j) = m - 1 : d_j = m - 2 \text{ (and } D_j = 1 - 2/m), \tag{8}$$

and in general

$$\text{For Max}(j) = m - h : d_j = m - 2h \text{ (and } D_j = 1 - 2h/m). \tag{9}$$

In short, we redefine the distance CurrentD used in the algorithm applied to a single solution x^* so that CurrentD represents the distance that a current solution has moved away from X^*. Then, starting from CurrentD $= 0$ for the solution x^S, flipping x_j^S to $1 - x_j^S$ results in

$$\text{CurrentD} := \text{CurrentD} + D_j. \tag{10}$$

Consequently, Phase 0 for X^* modifies Phase 0 for x^* simply by initializing $x^* = x^S$ and replacing CurrentD := CurrentD + 1 in the main routine by CurrentD := CurrentD + D_j.

For Phases 1 to 3, if $j \in N(\sim *)$, then, just as in the algorithm for a single solution x^*, the flip corresponds to moving away from X^* and hence in this case we let CurrentD := CurrentD − D_j.

Note that the diversification in Phase 0 that sets CurrentD := CurrentD + D_j implies that flipping variables x_j with larger D_j values move away from X^* more quickly (causing CurrentD to achieve a given value in fewer flips). The quickest departure from X^* would be to restrict flips to variables with $D_j = 1$, corresponding to selecting flips in the "intersection" of the solutions in X^*, which also corresponds to the situation of moving away from a single solution x^*. Thus, an option in Phase 0 for X^* is to bias the choice of x_j in favor of variables with larger D_j values (for example, by assigning probabilities based on the magnitude of D_j).

Again, we refer to the tabu search algorithm that initiates the search (and that is executed in Phase 3 with a different stopping criterion) as the basic algorithm.

Setting the stage for Phase 0 applied to X^*. Apply the basic algorithm to problem (P) until satisfying the initial stopping criterion or the Phase 3 stopping criterion (according to the stage executed) and update the set X^* to consist of a collection of elite solutions found. Identify the signature solution x^* for X^* and the associated values $D_j = d_j/m$ for each $j \in N$.

After the initial execution of the basic algorithm, choose an initial value FocalD as the minimum distance desired to move away from X^* and a value x_oThresh for the threshold value for moving away from x_o^S of the signature solution where x_o^S will be represented by x_o^* upon setting $x^* = x^S$. Also, select a value MaxFlip limiting the total number of flips executed in Phase 0.

Phase 0 for X^*.

Initialization Set $x^* = x^S$, $x^\# = x^*$ and CurrentD $= 0$. Once again, we define $N(*) = \{j \in N : x_j^\# = x_j^*\}$ and $N(\sim *) = \{j \in N : x_j^\# \neq x_j *\}$ and initialize $N(*) = N$ and $N(\sim *) = \emptyset$.

Main routine

While CurrentD < FocalD and $x_o^\# < x_o$Thresh;

  Choose $j \in N(*)$ and flip $x_j^\# (= x_j^*)$;

    $x_j^\# := 1 - x_j^\#$;

    $N(*) := N(*) \backslash \{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$;

    CurrentD := CurrentD + D_j;

    NumFlip := NumFlip + 1;

If (NumFlip = MaxFlip) exit the "While loop";
Endwhile

To launch subsequent phases, let FocalD = CurrentD, $x^L = x\text{Best} = x^{\#}$ and $x_o^L = x_o\text{Best} = x_o^{\#}$.

Phase 1 For X^* (quick descent to a local optimum satisfying CurrentD ⩾ FocalD).

Initialization For $N(*)$, $N(\sim *)$, CurrentD, FocalD and $x^{\#}$ inherited from Phase 0, let $x = x^{\#}$ and Stop = False.

Main routine

While Stop = False

1. Choose a highest evaluation 1-flip of a variable x_j, $j \in N$, by requiring that j satisfies $j \in N(*)$ or $j \in N(\sim *)$ and CurrentD $- D_j \geqslant$ FocalD.
   If this flip does not improve (decrease) x_o, then
   (a) Set Stop = True, set $x^L = x$ (and $x_o^L = x_o$) and proceed to Phase 2 with a local D-optimum x^L. Otherwise,
   (b) Execute the flip to produce a new solution, again denoted by x.
2. If $j \in N(*)$
   Let CurrentD := CurrentD + D_j and move j from $N(*)$ to $N(\sim *)$, giving $N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$.
   Else if $j \in N(\sim *)$
   Let CurrentD := CurrentD $- D_j$ and move j from $N(\sim *)$ to $N(*)$, giving $N(\sim *) := N(\sim *)\backslash\{j\}$ and $N(*) := N(*) \cup \{j\}$.

Endwhile

2-flip option. If the 1-flip in Step 1 above does not improve x_o, then choose a highest evaluation 2-flip of variables x_j and $x_k, j, k \in N$such that one of the following conditions holds: (i) $j, k \in N(*)$; (ii) $j \in N(*), k \in N(\sim *)$ and CurrentD$+D_j - D_k \geqslant$ FocalD or (iii) $j, k \in N(\sim *)$CurrentD$-D_j - D_k \geqslant$ FocalD. If this 2-flip also does not improve x_o then execute Step 1(a) above and otherwise execute Step 2 above for both j and k.

The 2-flip option can be examined before the 1-flip in Step 1 or they can both be examined simultaneously to pick the move that satisfies the "⩾ FocalD" requirement and improves x_o the most.

In anticipation of Phase 2, let $x\text{Best} = x^L$ and $x_o\text{Best} = x_o^L$.

Phase 2 For X^*.

Initialization Use the same initialization applied in the case where focal distance tabu search is applied to a single solution x^*, letting $x = x\text{Best}$, assigning a 0 initial tabu tenure to all x_j for $j \in N(*)$, and assigning a tabu tenure SmallTenure to all x_j for $j \in N(\sim *)$. If SmallTenure = 0, then all variables start non-tabu, and Phase 2 is skipped, passing directly to Phase 3.

Main routine

For Iter = 1 to LimIter
  Choose a highest evaluation non-tabu 1-flip of a variable $x_j, j \in N$.
  Option A: Require that j satisfies $j \in N(*)$ or $j \in N(\sim *)$ and CurrentD $- d_j/m \geqslant$ FocalD.
  Execute the flip to produce a new solution, again denoted by x.
  If $j \in N(*)$ let CurrentD := CurrentD + D_j and move j from $N(*)$ to $N(\sim *)$, giving
    $N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$.
  Else if $j \in N(\sim *)$ let CurrentD := CurrentD $- D_j$ and move j from $N(\sim *)$ to $N(*)$, giving
    $N(\sim *) := N(\sim *)\backslash\{j\}$ and $N(*) := N(*) \cup \{j\}$.
  If Option A is not applied and $j \in N(\sim *)$ above then perform the following:
  While CurrentD < FocalD
    Choose a highest evaluation non-tabu 1-flip of a variable $x_j, j \in N(*)$, set
    CurrentD := CurrentD + D_j and move j from $N(*)$ to $N(\sim *)$ giving
    $N(*) := N(*)\backslash\{j\}$ and $N(\sim *) := N(\sim *) \cup \{j\}$.

Endwhile
If $x_o < x_o$Best then update x_oBest $= x_o$ and xBest $= x$.
Endfor

As in Phase 1, additional variants are possible by including 2-flip moves.

Phase 3 for X^*. This phase is identical to Phase 3 when X^* consists of a single solution x^*.

The complete focal distance tabu search algorithm for X^* may then be summarized as follows, which differs in only a few (critical) ways from the summary for the single solution x^*. As before, the basic algorithm uses an initial stopping criterion that limits the number of iterations it executes when launching the method, and also uses a Phase 3 stopping criterion that limits the number of iterations it executes in Phase 3. The overall termination criterion for stopping the entire algorithm is based on the number of iterations performed that fail to improve the best solution x^* as well as being based on a lower limit on the value of x_oThresh and an upper limit on the value of FocalD.

Summary: focal distance tabu search algorithm for X^*.

Initial step: Apply the basic algorithm until satisfying the initial stopping criterion. Identify the set X^* of elite solutions and its signature solution x^S and select beginning x_oThresh and FocalD values for Phase 0.
While the overall termination criterion is not satisfied
Set Improve = True;
While Improve = True (Round Loop)
Carry out a Round consisting of parallel threads initiated by the current x^S and the current x_oThresh and FocalD values as follows:
Execute the diversification process of Phase 0 and update FocalD;
Execute the constrained local improvement algorithm of Phase 1;
Execute the abbreviated basic algorithm of Phase 2;
Execute the basic algorithm in Phase 3 until satisfying the Phase 3 stopping criterion.
Identify the best solution x^* obtained over all the threads and the value of FocalD that produced this x^*. If x^* is not improved over the x^* that initiated the current Round, then set Improve = False and terminate the Round Loop.
Otherwise, identify the set X^* that was produced by the basic algorithm to obtain this x^* and identify its signature solution x^S. Then continue the Round Loop.
Endwhile
Focal adjustment step.
If the overall stopping criterion is not satisfied, increase x_oThresh and the value of FocalD inherited from the Round Loop and continue by beginning with the X^* associated with the best solution x^* previously found.
Endwhile

As mentioned earlier, the initial step can apply the basic algorithm in either a single thread or in multiple threads. If multiple threads are used in this step, then X^* can be composed by comparing sets X^* of elite solutions from the different threads and selecting a subset consisting of the best ones. Tradeoffs within the execution of parallel threads, either in the Initial Step or afterward, can be considered by periodically querying the threads to identify superior x^* solutions produced to the current point, and re-allocating them to provide new starting solutions for threads that have produced poorer x^* solutions.

Tradeoffs in speed versus solution quality can be explored by options that jump over Phase 1 or Phase 2 and by not terminating the Round Loop until x^* has not improved for two or more iterations. Good rules for the initial stopping criterion and the Phase 3 stopping criterion can be determined by relaxed rules that allow the basic algorithm to run for a long time, and then observing when the solutions produced could have been obtained by stopping the algorithm earlier. This also applies to initially picking a large LimIter value for Phase 2 that can subsequently be reduced. Such experimentation may also discover a more effective rule for ending Phase 2 than relying on LimIter. There remains a possibility that earlier stopping points will still yield overall results that are as good, where solutions gradually improve to a desired level even if they are not as good at some intermediate point. It may additionally be observed that X^* may be constructed by reference to clustering, using clustering ideas as proposed by Glover and Laguna [16] and Samorani et al. [17]. For constructing X^*, the focal distance algorithm can keep track of the best solutions x^* (including in a variant that initially applies the single solution x^* approach) and then using clustering to group solutions that are closer together to form new X^* sets. Another possibility for exploiting clustering is to extract a "central" x^* in each cluster and use these points to compose a new X^*. Then the focal distance process of driving the solution away from this X^* would drive it away from all the central x^* solutions, and so would tend to drive the search into a region different from any seen before.

As noted in connection with the version of the method applied to a single solution x^*, the sequence from Phase 0 to Phase 2, potentially skipping over Phase 1 or Phase 2, can be executed several times in succession for increasing values of x_oThresh and FocalD, to identify values of these parameters that lead to interesting new solution outcomes before continuing to Phase 3. This option may be particularly relevant in the context of applying focal distance TS to a set X^*, although it may be useful to allow Phase 2 to run for additional iterations in order to determine the merit of particular values of the x_oThresh and FocalD parameters.

Finally, we note as in Remark 3 that for some kinds of problems a measure of CurrentD related to the Hamming distance can be inappropriate. The updates we use in the X^* case, which increment and decrement CurrentD by a value D_j that can differ from the Hamming distance increment $D_j = 1$, give a format for handling these problems where the Hamming distance is not suitable. In the general case, the moves employed may not be simple 1-flips or 2-flips but may involve operations such as insertions and exchanges of elements as in permutation and scheduling problems.

6 Conclusion

Focal distance tabu search can be applied in numerous ways, by selecting different trigger points for launching diversification and by choosing different values for the parameters of the algorithm, as well as by using different tabu search approaches such as those incorporating path relinking. A useful avenue for empirical research will be to identify what values of FocalD work better in different settings and what patterns work better for increasing this value each time a diversification step fails to provide an improvement in Phase 3, including the option of decreasing this value once Phase 3 succeeds in producing an improved x^* as well as of increasing the values of x_oThresh and FocalD in successive applications of the Phase 0 to Phase 2 sequence before proceeding to Phase 3. It will also be interesting to discover if the version based on a single x^* solution or the version based on set of solutions X^* works better, depending on the kind of problem encountered.

It is worth noting that the fundamental strategies in the initial step and in Phases 2 and 3 need not be restricted to implementation with tabu search, provided an algorithm is used that is capable of searching beyond a local optimum, though the adaptive memory and strategic oscillation components of tabu search reinforce several aspects of these steps.

The focal distance TS algorithm, both by reference to a single solution x^* and a set of solutions X^*, can be employed in a natural way in conjunction with a variety of metaheuristics that include diversification processes, including iterated local search as described in [18,19], and the adaptive perturbation procedures of breakout local search in [20–22]. Another potentially fertile application of the approach can be to augment the thresholding methods of late acceptance hill climbing in [23] and diversified late acceptance search in [24]. Joining the focal distance TS approach with GRASP, particularly in the versions of GRASP that incorporate path relinking as in [25], and with the dynamic diversification strategy in [26], afford additional opportunities for future research. Finally, focal distance TS can be used for analyzing and exploiting landscape patterns in combinatorial search as described in [27].

Acknowledgements We are grateful to the anonymous reviewers for their valuable comments which helped us to improve the paper.

References

1 Lucas A. Ising formulations of many NP problems. Front Phys, 2014, 2: 5
2 Kochenberger G, Hao J K, Glover F, et al. The unconstrained binary quadratic programming problem: a survey. J Comb Optim, 2014, 28: 58–81
3 Glover F, Kochenberger G, Du Y. Quantum bridge analytics I: a tutorial on formulating and using QUBO models. 4OR-Q J Oper Res, 2019, 17: 335–371
4 Vyskocil T, Pakin S, Djidjev H N. Embedding inequality constraints for quantum annealling optimization. In: Quantum Technology and Optimization Problems. Cham: Springer, 2019. 11413
5 Glover F. Future paths for integer programming and links to artificial intelligence. Comput Oper Res, 1986, 13: 533–549
6 Glover F. Heuristics for integer programming using surrogate constraints. Decision Sci, 1977, 8: 156–166

7 Dueck G, Scheuer T. Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J Comput Phys, 1990, 90: 161–175

8 Osman I H. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. Ann Oper Res, 1993, 41: 421–451

9 Dueck G. New optimization heuristics. the great deluge algorithm and the record-to-record travel. J Comput Phys, 1993, 104: 86–92

10 Glover F. Tabu thresholding: improved search by nonmonotonic trajectories. ORSA J Computing, 1995, 7: 426–442

11 Glover F, Kochenberger G A. Critical event tabu search for multidimensional knapsack problems. In: Meta-Heuristics: Theory & Applications. Boston: Springer, 1996. 407–427

12 Battiti R, Tecchiolli G. The reactive tabu search. ORSA J Comput, 1994, 6: 126–140

13 Kelly J P, Laguna M, Glover F. A study of diversification strategies for the quadratic assignment problem. Comput Oper Res, 1994, 21: 885–893

14 Woodruff D L, Zemel E. Hashing vectors for tabu search. Ann Oper Res, 1993, 41: 123–137

15 Glover F, Kochenberger G, Alidaee B, et al. Tabu search with critical event memory: an enhanced application for binary quadratic programs. In: Meta-Heuristics. Boston: Springer, 1998. 83–109

16 Glover F, Laguna M. Tabu Search. Boston: Springer Academic Publications, 1997

17 Samorani M, Wang Y, Wang Y, et al. Clustering-driven evolutionary algorithms: an application of path relinking to the quadratic unconstrained binary optimization problem. J Heuristics, 2019, 25: 629–642

18 Lourenço H R, Martin O C, Stüzle T. Iterated local search: framework and applications. In: Handbook of Metaheuristics. Boston: Springer, 363–397

19 Coelho V N, Grasas A, Ramalhinho H, et al. An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. Eur J Oper Res, 2016, 250: 367–376

20 Benlic U, Hao J K. Breakout local search for the quadratic assignment problem. Appl Math Comput, 2013, 219: 4800–4815

21 Fu Z H, Hao J K. Breakout local search for the Steiner tree problem with revenue, budget and hop constraints. Eur J Oper Res, 2014, 232: 209–220

22 Lu Z, Hao J K, Zhou Y. Stagnation-aware breakout tabu search for the minimum conductance graph partitioning problem. Comput Oper Res, 2019, 111: 43–57

23 Burke E K, Bykov Y. The late acceptance hill-climbing heuristic. Eur J Oper Res, 2017, 258: 70–78

24 Namazi M, Sanderson C, Newton M A H, et al. Diversified late acceptance search. In: Advances in Artificial Intelligence. Cham: Springer, 2018, 11320: 299–311

25 Resende M G C, Ribeiro C C. Greedy randomized adaptive search procedures: advances and extensions. In: Handbook of Metaheuristics. 3rd ed. Cham: Springer, 2019. 169–220

26 Peng B, Liu D, Su Z, et al. Solving the Incremental Graph Drawing Problem by Solution-based Multiple Neighborhood Tabu Search. Research Report. Chengdu: Southwestern University of Finance and Economics, 2019

27 Glover F. Analyzing and Exploiting Landscape Patterns in Combinatorial Optimization. School of Engineering and Applied Science, University of Colorado. 2019