Discrete Applied Mathematics 308 (2022) 235–254

Contents lists available at ScienceDirect

Discrete Applied Mathematics

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

On convergence of scatter search and star paths with directional rounding for 0–1 mixed integer programs

Raca Todosijević [a], Saïd Hanafi [a,*], Fred Glover [b]

[a] LAMIH UMR 8201 CNRS, Université Polythenique Hauts de France, Valenciennes, France
[b] OptTek Systems, Inc., 2241 17th Street, Boulder, CO 80302, USA

ARTICLE INFO

Article history:
Received 4 December 2018
Received in revised form 6 March 2020
Accepted 7 May 2020
Available online 13 June 2020

Keywords:
Metaheuristics
Convergence
Scatter search
Star path
Directional rounding

ABSTRACT

Scatter Search is an evolutionary metaheuristic introduced by Glover (1977) as a heuristic for integer programming and was joined with a directional rounding strategy for 0–1 Mixed Integer Programming (MIP) problems based on Star Paths in Glover (1995). In this paper, we address directional rounding both independently and together with these other algorithmic components, studying its properties as a mapping from continuous to discrete (binary) space. We establish several useful properties of directional rounding and show that it provides an extension of classical rounding and complementing operators. Moreover, we observe that directional rounding of a line, as embodied in a Star Path, contains a finite number of distinct 0–1 points. This property, together with those of the solution space of 0–1 MIP, enables us to organize the search for an optimal solution of 0–1 MIP problems using Scatter Search in association with both cutting plane and extreme point solution approaches. Building on this, we provide a Convergent Scatter Search algorithm for 0–1 Mixed Integer Programs with proof of its finite convergence, along with two variants of its implementation and examples that illustrate the running of the approach.

1. Introduction

Zero–one (binary) mixed integer programming is used for modeling many combinatorial problems, ranging from logical design to scheduling and routing as well as encompassing graph theory models for resource allocation and financial planning. A 0–1 mixed integer program (MIP) may be written in the following form:

$$\begin{array}{ll} \textit{minimize} & z = cx \\ s.t. & Ax = d \\ & 0 \le x_j \le U_j,\ j \in N = \{1, \dots, n\} \\ & x_j \in \{0, 1\},\ j \in I \subseteq N \end{array} \tag{1}$$

A is a constant matrix, d is a constant vector, the set N denotes the index set of variables, while the set I contains indices of binary variables. Each variable x_j has an upper bound denoted by U_j (which equals 1 if x_j is binary, as in the present case for the integer variables), and otherwise may be infinite. It is assumed that all continuous variables can be represented (either directly or by transformation) as slack variables, i.e., the associated columns of the (possibly transformed) matrix

* Corresponding author.
E-mail addresses: racatodosijevic@gmail.com, Raca.Todosijevic@uphf.fr (R. Todosijević), Said.Hanafi@uphf.fr (S. Hanafi), glover@opttek.com (F. Glover).

https://doi.org/10.1016/j.dam.2020.05.010

A constitute an identity matrix. Hence, if the values of binary variables are known, the continuous variables receive their values automatically. The integer problem defined in this manner will be denoted simply by IP while the relaxation of IP obtained by excluding integrality constraints will be denoted by LP. A feasible solution of $IP(LP)$ will be called $IP(LP)$ feasible.

The Scatter Search evolutionary metaheuristic combines decision rules and problem constraints, and it has its origins in surrogate constraint strategies. Scatter Search, unlike Genetic Algorithms, operates on a small set of solutions and makes only limited use of randomization as a proxy for diversification when searching for a globally optimal solution. Since its introduction in [16,18] as a heuristic for integer programming, Scatter Search has been successfully applied to a wide range of combinatorial optimization problems (see e.g., [26,32,39,41] for recent applications). The basic scatter search design, which can be implemented in varying degrees of sophistication, can be expressed in terms of a five method template [20,21,24,25,33,34,40,42]:

(i) A Diversification Generation Method to generate a collection of diverse trial solutions within the search space.

(ii) An Improvement Method to transform a trial solution into one or more enhanced trial solutions.

(iii) A Reference Set Update Method to build and maintain a reference set consisting of the b best solutions found, where the value of b is typically small e.g. no more than 20. Solutions gain membership in the reference set according to their quality or their diversity.

(iv) A Subset Generation Method to operate on the reference set, to produce several subsets of its solutions as a basis for creating combined solutions.

(v) A Solution Combination Method to transform a given subset of solutions produced by the Subset Generation Method into one or more combined solution vectors.

Scatter search and its Path Relinking generalization have been successfully applied in a wide range of discrete and nonlinear optimization settings including neural networks, routing problems, graph drawing, scheduling, linear ordering, assignment, p–Median, knapsack, coloring problems, clustering/selection and software testing (see for example [42]).

In the literature, there are more than dozen heuristics for 0–1 Mixed Integer Programs. Some of them are concerned with finding a first MIP feasible solution which is NP hard problem by itself. Among the better known heuristics of this type are the Feasibility Pump heuristics [2,9,14], the Variable Neighborhood Pump approach [28], and the Single and Variable Neighborhood Diving heuristics [37]. The second class consists of improvement procedures for 0–1 MIP that require a feasible MIP solution as input. Heuristics of this type include Local Branching [15], Variable Neighborhood Branching [31], and Relaxation Induced Neighborhood Search (RINS) [11]. The last class of heuristics are those that neither require a feasible MIP solution as input nor finish their work after finding the first feasible solution. Prominent members of this group include OCTAhedral Neighbourhood Enumeration (OCTANE) [5], Pivot and Complement [1,7], Pivot and Shift [8], Tabu Search [38], Pivot-Cut-and-Dive [13] and various Matheuristics [29,30,36,45].

The introduction of Star Paths with directional rounding for 0–1 Mixed Integer Program as a supporting strategy for Scatter Search in [19] established basic properties of directional rounding and provided efficient methods for exploiting them. The most important of these properties is the existence of a plane (which can be required to be a valid cutting plane for IP) which contains a point that can be directionally rounded to yield an optimal solution and which, in addition, contains a convex subregion all of whose points directionally round to give this optimal solution. Several alternatives are given in [19] for creating such a plane as well as a procedure to explore it using principles of Scatter Search. This work also shows that the set of all 0–1 solutions obtained by directionally rounding points of a given line (the so-called Star Path) contains a finite number of different 0–1 solutions and provides a method to generate these solutions efficiently. Glover and Laguna [23] elaborate these ideas and extend them to General Mixed Integer Programs by means of a more general definition of directional rounding.

The main contributions of this paper are:

- We start from the work of Glover (1995) [19] where directional rounding is introduced. In this paper, we prove the theorems in Glover's paper, which were stated but not proved, and also provide some corrections and extensions of that work. We additionally identify new properties (for example, showing that the directional rounding operator provides an extension of the complement and rounding operators). Finally, we describe connections between continuous and discrete solution space and introduce a means to project from continuous to discrete space (for example, by rounding in relation to a cone), and introduce new theorems that underlie these connections.
- The main contribution of our work is to propose new convergent algorithms for 0–1 mixed integer programming within a framework that joins scatter search with the methodology of star paths and directional rounding. This marriage allows us to prove finite convergence of scatter search for the first time, and thus to address a gap in the literature on metaheuristics which to date has chiefly focused on applied methods with little attention to theory.
- To reinforce our theoretical contributions we also propose two implementations of the convergent scatter search algorithm with an illustrative example.

We additionally point out that the proposed convergent algorithms use the main components of scatter search since the directional rounding with star paths uses the methodology of combining solutions by linear combination and the polyhedron generated can be considered as the set of reference solutions employed in scatter search. Finally, as in scatter search we propose a local search (in this case a special form) which is designed to improve the projected solution. Note that our proposal can also be used to generate diversity in the population, as advocated in scatter search.

The rest of the paper is organized as follows. Section 2 begins by proving key properties of directional rounding and provides an efficient method for directional rounding of a line, and hence for generating Star Paths. In Section 3, we state and prove several theorems which enable us to organize search for an optimal solution of 0–1 MIP problems using Scatter Search. In Section 4, we propose Convergent Scatter Search Algorithms and we provide the proof of their finite convergence. Additionally, we give examples to illustrate the execution of those Convergent Scatter Search Algorithms based on directional rounding. Finally, Section 5 concludes the paper.

2. Generating star paths with directional rounding

In this section we introduce the notion of directional rounding and present its key features. In addition, we show how directional rounding may be efficiently used to project from continuous to discrete (binary) space. In particular, we demonstrate how directional rounding of a line may be performed in an efficient way. As a foundation for presenting these results, we first introduce basic notations that will be used throughout the paper.

2.1. Basic notations

Throughout the paper we assume that each point x belongs to $\mathbb{R}^n$, unless stated otherwise. For two arbitrary points $x' \in \mathbb{R}^n$ and $x'' \in \mathbb{R}^n$, we identify:
the ray from x' through x'' by

$$Ray(x', x'') = \{x' + \lambda(x'' - x') : \lambda \geq 0\};$$

the line joining x' and x'' by

$$Line(x', x'') = \{x' + \lambda(x'' - x') : \lambda \in \mathbb{R}\};$$

and the segment with extremities x' and x'' by

$$[x', x''] = \{x' + \lambda(x'' - x') : 0 \leq \lambda \leq 1\}.$$

Let $X(R)$ denote a chosen set of reference points, indexed by the set R i.e., $X(R) = \{x(r) : r \in R\}$. Since, the points of $X(R)$ are linearly independent in the usual case to be considered, we define the hyperplane consisting of all normalized linear combinations of these points by:

$$Plane(X(R)) = \{\sum_{r\in R} \lambda_r x(r) : \sum_{r\in R} \lambda_r = 1\}. \tag{2}$$

Furthermore, we identify the associated half space as:

$$Half\_space(X(R)) = \{\sum_{r\in R} \lambda_r x(r) : \sum_{r\in R} \lambda_r \geq 1\}. \tag{3}$$

If we choose a point x^* which does not belong to the $Plane(X(R))$ and which therefore constitutes a set of affinely independent points together with points from $X(R)$, we will be able to define the polyhedral (half) cone spanned by the rays from x^* through the points of X(R):

$$Cone(x^*, X(R)) = \{x^* + \sum_{r\in R} \lambda_r (x(r) - x^*) : \lambda_r \geq 0, r \in R\}. \tag{4}$$

Finally, the intersection of $Cone(x^*, X(R))$ with $Half\_space(X(R))$, i.e., the face of the truncated cone that excludes the point x^* is defined as:

$$Face(X(R)) = \{\sum_{r\in R} \lambda_r x(r) : \sum_{r\in R} \lambda_r = 1,\ \lambda_r \geq 0, r \in R\}. \tag{5}$$

The set in $\mathbb{R}^n$ is said to be polyhedral if it is the intersection of a finite number of closed half spaces. A polytope is a polyhedral set which is bounded. In addition, a polytope is a convex hull of a finite set of points.

The Hamming distance between two solutions, $x' \in \{0, 1\}^n$ and $x \in [0, 1]^n$ can be represented as:

$$d(x', x) = \textstyle\sum_{j=1}^{n} |x'_j - x_j| = x(e - x') + x'(e - x), \tag{6}$$

where $e = (1, 1, \ldots, 1)$ is the vector of all 1's with appropriate dimension.

Rounding and complementing operators are useful in mathematical programming with 0–1 variables. We will show that the directional rounding introduced by Glover [19] provides an extension of these operators.

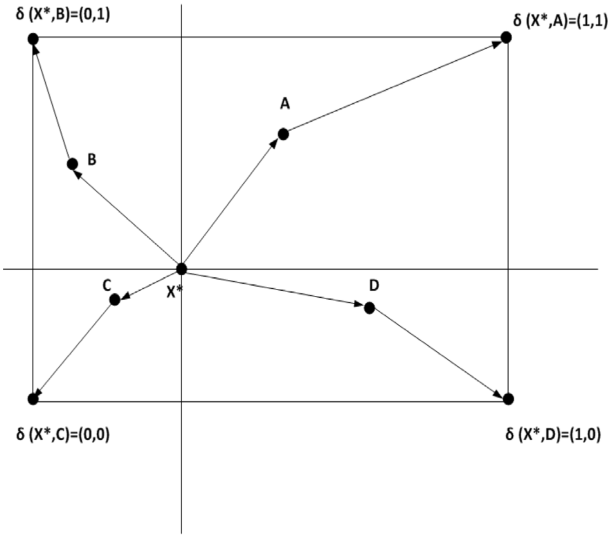

Fig. 1. Example of directional rounding in two-dimensional case.

2.2. Definition and properties of directional rounding

Directional rounding is a mapping δ from the continuous space $[0, 1]^n$ to the discrete space $\{0, 1\}^n$ by the following rules. The directional rounding $\delta(x^*, x')$, from a base point $x^* \in [0, 1]^n$ to an arbitrary focal point x', is the point in $\{0, 1\}^n$ given by

$$\delta(x^*, x') = (\delta(x_j^*, x_j') : j \in N)$$

where $\delta(x_j^*, x_j')$ is defined as

$$\delta(x_j^*, x_j') = \begin{cases} 0 & \text{if } x_j' < x_j^* \\ 1 & \text{if } x_j' > x_j^* \\ x_j^* & \text{if } x_j' = x_j^* \in \{0, 1\} \\ 0 \text{ or } 1 & \text{if } x_j' = x_j^* \notin \{0, 1\} \end{cases} \tag{7}$$

It may be noted that directional rounding includes nearest neighbor rounding as a special instance, i.e., $[x_j'] = \delta(0.5, x_j')$. The directional rounding in the last case $x_j' = x_j^* \notin \{0, 1\}$ can be performed in several ways. Some of them are:

- use simple rounding, i.e., $\delta(x_j^*, x_j^*) = [x_j^*]$ breaking the tie arbitrarily for $x_j^* = 0.5$,
- choose 0 or 1 randomly
- make the choice according to values of x_j^* and x_j' in the following way. If $x_j' \leq 1/3$ or $2/3 \leq x_j' \leq 1$ perform simple rounding, otherwise choose 0 or 1 randomly
- choose at random a segment $[y', y'']$ which contains x^* and after that set $\delta(x_j^*, x_j^*) = 1$ if $y_j' < y_j''$ while otherwise set $\delta(x_j^*, x_j^*) = 0$.

Fig. 1 illustrates directional rounding for different focal points labeled A, B, C, and D.

We define the directional rounding $\delta(x^*, X)$ from the point x^* to the set X to be the set of points in $\{0, 1\}^n$ given by

$$\delta(x^*, X) = \{\delta(x^*, x') : x' \in X\}.$$

The motivation for directional rounding comes from the fact that if x^* is an extreme point of the feasible set of LP relaxation, then every feasible 0–1 solution can be obtained by directional rounding relative to focal points that lie within the LP cone defined by non-negative values for the current nonbasic variables (as proved in Section 3).

The next two lemmas consider special cases where either the base point or the focal point is a binary vector. More precisely, Lemma 2.1 states that any vertex of the unit hypercube $x' \in \{0, 1\}^n$ is the fixed point of directional rounding.

Lemma 2.1. *For any base point $x^* \in [0, 1]^n$ and focal point $x' \in \{0, 1\}^n$*

$$\delta(x^*, x') = x'.$$

Proof. From the definition of directional rounding, for all $j \in N$ we have $\delta(x_j^*, 1) = 1$ and $\delta(x_j^*, 0) = 0$, hence $\delta(x_j^*, x_j') = x_j'$ for $x_j' \in \{0, 1\}$. Hence, $\delta(x^*, x') = x'$ for each point $x' \in \{0, 1\}^n$. □

Lemma 2.2 states that directional rounding of any point x' relative to the base point $x^* \in \{0, 1\}^n$ yields point deduced from x^* by complementing only those coordinates that have different values in x^* and x'.

Lemma 2.2. *For any base point $x^* \in \{0, 1\}^n$ and focal point $x' \in [0, 1]^n$*

$$\delta(x_j^*, x_j') = \begin{cases} 1 - x_j^* \text{ if } x_j' \neq x_j^* \\ x_j^* \text{ otherwise} \end{cases} \tag{8}$$

Proof. If $x_j' < x_j^*$ then x_j^* must equal 1, so $\delta(x_j^*, x_j') = 0$, which is equal to $1 - x_j^*$. Similarly, if $x_j' > x_j^*$ then x_j^* must equal 0 and therefore $\delta(x_j^*, x_j') = 1 - x_j^*$. Finally, in the case $x_j' = x_j^*$ the definition of directional rounding for the case $x_j' = x_j^* \in \{0, 1\}^n$ gives $\delta(x_j^*, x_j') = x_j^*$. □

As a consequence of Lemma 2.2, it follows that directional rounding generalizes the standard complementing operator as stated in the following corollary.

Corollary 2.3. *For any base point $x^* \in \{0, 1\}^n$ and focal point $x' \in]0, 1[^n$*

$$\delta(x^*, x') = e - x^*.$$

The next lemma says that every point on the ray from x^* through x' (excluding the point x^* itself) gives the same directional rounding as $\delta(x^*, x')$.

Lemma 2.4. *For any base point $x^* \in [0, 1]^n$ and a focal point $x' \neq x^*$*

$$\delta(x^*, x') = \delta(x^*, x'')$$

for all $x'' \in Ray(x^*, x')$ such that $x'' \neq x^*$.

Proof. Since $x'' \in Ray(x^*, x')$ there exists $\lambda \geq 0$ such that $x'' = \lambda x' + (1 - \lambda)x^*$. Therefore for each component x_j'', $j \in N$, we have three cases:

$$\begin{cases} (i)\ x_j' < x_j^* \ \Rightarrow x_j'' < x_j^*, \\ (ii)\ x_j' > x_j^* \ \Rightarrow x_j'' > x_j^*, \\ (iii)\ x_j' = x_j^* \ \Rightarrow x_j'' = x_j^*, \end{cases} \tag{9}$$

These observations lead to the conclusion that $\delta(x_j^*, x_j') = \delta(x_j^*, x_j'')$ for each $j \in N$ which further implies $\delta(x^*, x') = \delta(x^*, x'')$. □

2.3. Efficient procedure for generating a Star Path

A Star Path $L(x^*, x', x'')$ is defined as a set of 0–1 vectors obtained by directional rounding using points which belong to the line connecting x' and x'' as focal points and x^* as a base point. More precisely,

$$L(x^*, x', x'') = \delta(x^*, Line(x', x'')).$$

In this section, we provide several properties which lead to an efficient procedure to compute $\delta(x^*, S)$ where S is a subset of $Line(x', x'')$.

Lemma 2.5. *Given x' and x'' as focal points and x^* as a base point, define $\lambda_j^* = \frac{x_j^* - x_j'}{x_j'' - x_j'}$ for $j \in N^{\neq} = \{j \in N : x_j'' \neq x_j'\}$. For every real value of λ, the elements of the vector $\delta(x^*, x' + \lambda(x'' - x'))$ in $L(x^*, x', x'')$ are given by*

$$\delta(x_j^*, x_j' + \lambda(x_j'' - x_j')) = \begin{cases} \delta(x_j^*, x_j') & \text{if } x_j'' = x_j' \\ 1 & \text{if } (\lambda < \lambda_j^* \text{ and } x_j'' < x_j') \text{ or } (\lambda > \lambda_j^* \text{ and } x_j'' > x_j') \\ 0 & \text{if } (\lambda < \lambda_j^* \text{ and } x_j'' > x_j') \text{ or } (\lambda > \lambda_j^* \text{ and } x_j'' < x_j') \\ \delta(x_j^*, x_j^*) & \text{if } (x_j'' \neq x_j') \text{ and } (\lambda = \lambda_j^*) \end{cases} \tag{10}$$

Proof. Let $x_j = x_j' + \lambda(x_j'' - x_j')$. The case where $x_j' = x_j''$ is obvious. For the case where $x_j'' > x_j'$, the condition $\lambda_j^* < \lambda$ implies

$$\lambda_j^* = \frac{x_j^* - x_j'}{x_j'' - x_j'} < \lambda = \frac{x_j - x_j'}{x_j'' - x_j'}.$$

Hence $x_j^* < x_j$ which means that $\delta(x_j^*, x_j) = 1$, and the condition $\lambda_j^* > \lambda$ implies $x_j^* > x_j$ which means that $\delta(x_j^*, x_j) = 0$. Similarly, in the case $x_j'' < x_j'$, we have $\delta(x_j^*, x_j) = 1$ if $\lambda_j^* > \lambda$ and $\delta(x_j^*, x_j) = 0$ if $\lambda_j^* < \lambda$. In the last case where $\lambda = \lambda_j^*$ we have $x_j = x_j^*$. □

The definition of the element $\delta(x_j^*, x_j' + \lambda(x_j'' - x_j'))$ needs a specific "tie breaking" rule to handle the case $\lambda = \lambda_j^*$ (which corresponds to $\delta(x_j^*, x_j' + \lambda(x_j'' - x_j') = \delta(x_j^*, x_j^*))$, where the original definition of directional rounding requires such a rule to choose between a value of 0 or 1, in our implementation, we use simple rounding. The proof of the preceding Lemma 2.5 does not depend on the restriction on λ.

Let $(\pi(1), \pi(2), \ldots, \pi(t))$ be a permutation of the indexes of $N^{\neq} = \{j \in N : x_j'' \neq x_j'\}$ so that $\lambda_{\pi(1)}^* \leq \lambda_{\pi(2)}^* \leq \cdots \leq \lambda_{\pi(t)}^*$, where $t = |N^{\neq}|$. We will assume that the $\lambda_{\pi(h)}^*$ values are all distinct so that $\lambda_{\pi(h)}^* < \lambda_{\pi(h+1)}^*$ for all $h < t$.

Theorem 2.6. *Given x' and x'' as focal points and x^* as a base point, let $\lambda_j^* = \frac{x_j^* - x_j'}{x_j'' - x_j'}$ for $j \in N^{\neq}$ and define $l(\lambda) = \delta(x^*, x' + \lambda(x'' - x'))$. Then for each $h = 1, 2, \ldots, |N^{\neq}| - 1$ and for every λ and λ' such that $\lambda, \lambda' \in]\lambda_{\pi(h)}^*, \lambda_{\pi(h+1)}^*[$, we have*

$$l(\lambda) = l(\lambda').$$

Moreover, $d(l(\lambda_{\pi(h+1)}^*), l(\lambda_{\pi(h)}^*)) \leq 2$, and more precisely

$$l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*) \text{ for } j \in N - \{\pi(h), \pi(h+1)\}, \tag{11}$$

$$l_{\pi(h)}(\lambda_{\pi(h+1)}^*) = \begin{cases} 0 & \text{if } x_{\pi(h)}'' < x_{\pi(h)}' \\ 1 & \text{if } x_{\pi(h)}'' > x_{\pi(h)}' \end{cases} \tag{12}$$

$$l_{\pi(h+1)}(\lambda_{\pi(h+1)}^*) = \delta(x_{\pi(h+1)}^*, x_{\pi(h+1)}^*). \tag{13}$$

Proof. According to Lemma 2.5, the value $l_j(\lambda)$ for $j \in N^{=} = \{j \in N : x_j'' = x_j'\}$ does not depend on λ and is the same for all points from the line connecting points x' and x''. Hence $l_j(\lambda) = l_j(\lambda')$ for $j \in N^{=}$. From the assumption that $\lambda_{\pi(h)}^* < \lambda_{\pi(h+1)}^*$ for all $h < |N^{\neq}|$, we can partition the set $N^{\neq}$ into two sets $N' = \{j \in N^{\neq} : \lambda_j^* \leq \lambda_{\pi(h)}^*\}$ and $N'' = \{j \in N^{\neq} : \lambda_j^* \geq \lambda_{\pi(h+1)}^*\}$. For $j \in N'$ implies $\lambda_j^* < \lambda$, and $\lambda_j^* < \lambda'$ hence by Lemma 2.5, $l_j(\lambda) = l_j(\lambda')$. Similarly, $j \in N''$ implies $\lambda_j^* \geq \lambda_{\pi(h+1)}^* > \lambda > \lambda_{\pi(h)}^*$, and $\lambda' > \lambda_{\pi(h)}^*$ and hence holds $l_j(\lambda) = l_j(\lambda')$. Hence, $l_j(\lambda) = l_j(\lambda')$ for each $j \in N^{\neq}$. Similarly for $j \in N^{\neq} - \{\pi(h), \pi(h+1)\}$ it follows that

$$l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*).$$

Indeed, for each $j \in N^{\neq} - \{\pi(h), \pi(h+1)\}$ such that $\lambda_j^* < \lambda_{\pi(h)}^* < \lambda_{\pi(h+1)}^*$ we have $l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*) = 1$ if $x_j'' > x_j'$ and $l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*) = 0$ otherwise. Likewise for each $j \in N^{\neq} - \{\pi(h), \pi(h+1)\}$ such that $\lambda_j^* > \lambda_{\pi(h+1)}^* > \lambda_{\pi(h)}^*$ we obtain $l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*)$. Finally, the formulas (12) and (13) follow directly the definition of δ (see Eq. (19)). □

The preceding theorem reveals that directional rounding of a line may be performed in a finite number of steps since directional rounding of any point belonging to the open line segment $]x' + \lambda_{\pi(h)}^*(x'' - x'), x' + \lambda_{\pi(h+1)}^*(x'' - x')[$ will map to the same 0–1 point. In addition, it shows that the points $l_{\pi(h+1)}$ and $l_{\pi(h)}$ differ in at most two coordinates. We show in the following example that the distance $d(l(\lambda_{\pi(h+1)}^*), l(\lambda_{\pi(h)}^*))$ can be equal to 2.

Example 2.7. Consider the following simple example in two dimensions where the base point $x^* = (0, 1)$ and the focal points are given by $x' = (0.1, 0.8)$ and $x'' = (\frac{2}{30}, 0.6)$, with the corresponding values $\lambda_1^* = 3$ and $\lambda_2^* = -1$. It is easy to see that $l(\lambda_1^*) = (0, 0)$ and $l(\lambda_2^*) = (1, 1)$. Hence, the Hamming distance between the two vectors $l(\lambda_1^*)$ and $l(\lambda_2^*)$ is equal to 2.

Example 2.8. In contrast to the preceding example, suppose the base point is $x^* = (0.3, 0.4)$ and the focal points are given by $x' = (0.2, 0.3)$ and $x'' = (0.4, 0.7)$. Then, the values of λ_1^* and λ_2^* are $\lambda_1^* = 0.5$, $\lambda_2^* = 0.25$. Hence it follows that $l(\lambda_1^*) = (0, 1)$ and $l(\lambda_2^*) = (1, 1)$, and in this case the Hamming distance between the two vectors $l(\lambda_1^*)$ and $l(\lambda_2^*)$ is equal to 1.

In what follows we will show that under stronger assumptions the distance between two consecutive points $l(\lambda_{\pi(h)}^*)$ and $l(\lambda_{\pi(h+1)}^*)$ is equal to 1. As a foundation, we prove the following lemma.

Lemma 2.9. *The assumption $0 < \lambda_j^* < 1$, $j \in N$ is equivalent to the condition that the element x_j^* is on the open segment between the elements x_j' and x_j'', i.e., $\min\{x_j', x_j''\} < x_j^* < \max\{x_j', x_j''\}$.*

Proof. Since $\lambda_j^*, j \in N$ is defined only if $x'_j \neq x''_j$, we consider the following two cases.
Case 1: $(x''_j > x'_j) \Leftrightarrow (x''_j - x'_j) > 0$.
Using the definition of λ_j^* and assumption $0 < \lambda_j^* < 1$ we obtain the following set of equivalences,

$$\lambda_j^* < 1 \Leftrightarrow \frac{x_j^* - x'_j}{x''_j - x'_j} < 1 \Leftrightarrow (x_j^* - x'_j) < (x''_j - x'_j) \Leftrightarrow x_j^* < x''_j$$

$$\lambda_j^* > 0 \Leftrightarrow \frac{x_j^* - x'_j}{x''_j - x'_j} > 0 \Leftrightarrow (x_j^* - x'_j) > 0 \Leftrightarrow x_j^* > x'_j$$

which lead to the conclusion that

$$(x''_j > x'_j) \text{ and } (0 < \lambda_j^* < 1) \Leftrightarrow x'_j < x_j^* < x''_j.$$

Case 2: $(x''_j < x'_j) \Leftrightarrow (x''_j - x'_j) < 0$.
Using the same facts as in the previous case, we obtain the following relations

$$\lambda_j^* < 1 \Leftrightarrow \frac{x_j^* - x'_j}{x''_j - x'_j} < 1 \Leftrightarrow (x_j^* - x'_j) > (x''_j - x'_j) \Leftrightarrow x_j^* > x''_j$$

$$\lambda_j^* > 0 \Leftrightarrow \frac{x_j^* - x'_j}{x''_j - x'_j} > 0 \Leftrightarrow (x_j^* - x'_j) < 0 \Leftrightarrow x_j^* < x'_j$$

which imply

$$(x''_j < x'_j) \text{ and } (0 < \lambda_j^* < 1) \Leftrightarrow x''_j < x_j^* < x'_j$$

Finally, the conclusions derived in these two cases imply that the condition $0 < \lambda_j^* < 1$ is equivalent to the condition that x_j^* is on the segment between the elements x'_j and x''_j. □

Corollary 2.10. *Given x' and x'' as focal points and x^* as a base point, let $\lambda_j^* = \frac{x_j^* - x'_j}{x''_j - x'_j}$ for $j \in N^{\neq}$. Assume $0 < \lambda_{\pi(1)}^* < \lambda_{\pi(2)}^*, \dots, < \lambda_{\pi(t)}^* < 1$ with $t = |N^{\neq}|$, and suppose further that ties in directional rounding $\delta(x_j^*, x_j^*)$ which arises in the case $0 < x_j^* < 1$ are broken in the following way*

$$\delta(x_j^*, x_j^*) = \begin{cases} 0 & \text{if } x''_j < x'_j \\ 1 & \text{if } x''_j > x'_j \end{cases} \tag{14}$$

Then the distance between vectors $l(\lambda_{\pi(h)}^*)$ and $l(\lambda_{\pi(h+1)}^*)$ satisfies

$$d(l(\lambda_{\pi(h)}^*), l(\lambda_{\pi(h+1)}^*)) = 1.$$

More precisely,

$$l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*) \text{ for } j \in N - \{\pi(h+1)\}$$

$$l_{\pi(h+1)}(\lambda_{\pi(h+1)}^*) = 1 - l_{\pi(h+1)}(\lambda_{\pi(h)}^*).$$

Proof. The condition $0 < \lambda_{\pi(1)}^* < \lambda_{\pi(2)}^*, \dots, < \lambda_{\pi(t)}^* < 1$ implies $0 < x_{\pi(i)}^* < 1, i = 1, \dots t$, because as previously shown (Lemma 2.9) the condition $0 < \lambda_j^* < 1$ compels x_j^* to belong to the segment between x''_j and x'_j. Hence, $l_j(\lambda)$ may be rewritten using its definition in Lemma 2.5 and the tie breaking rule (14) as:

$$l_j(\lambda) = \begin{cases} \delta(x_j^*, x'_j) & \text{if } x'_j = x''_j \\ 0 & \text{if } (\lambda \geq \lambda_j^* \text{ and } x''_j < x'_j) \text{ or } (\lambda < \lambda_j^* \text{ and } x''_j > x'_j) \\ 1 & \text{if } (\lambda \geq \lambda_j^* \text{ and } x''_j > x'_j) \text{ or } (\lambda < \lambda_j^* \text{ and } x''_j < x'_j) \end{cases} \tag{15}$$

Therefore, $l_{\pi(h)}(\lambda_{\pi(h)}^*)$ equals 0 if $x''_{\pi(h)} < x'_{\pi(h)}$, and equals 1 if $x''_{\pi(h)} > x'_{\pi(h)}$. On the other hand, $l_{\pi(h)}(\lambda_{\pi(h+1)}^*)$ equals 0 if $x''_{\pi(h)} < x'_{\pi(h)}$, and otherwise, equals 1. Therefore, $l_{\pi(h)}(\lambda_{\pi(h)}^*) = l_{\pi(h)}(\lambda_{\pi(h+1)}^*)$. Further, $l_{\pi(h+1)}(\lambda_{\pi(h)}^*)$ equals 0 if $x''_{\pi(h+1)} > x'_{\pi(h+1)}$, and otherwise it equals 1, while $l_{\pi(h+1)}(\lambda_{\pi(h+1)}^*)$ equals 1 if $x''_{\pi(h+1)} > x'_{\pi(h+1)}$ and otherwise, it equals 0. Hence,

$$l_{\pi(h+1)}(\lambda_{\pi(h+1)}^*) = 1 - l_{\pi(h+1)}(\lambda_{\pi(h)}^*).$$

These conclusions together with Theorem 2.6 demonstrate that $l_j(\lambda_{\pi(h+1)}^*) = l_j(\lambda_{\pi(h)}^*) \text{ for } j \in N - \{\pi(h+1)\}$. □

We show that the condition $\lambda_{\pi(1)}^* = min\{\lambda_j^* : j \in N^{\neq}\} > 0$ is necessary in the previous corollary. Consider the following example.

Example 2.11. Let $x^* = (1, 0.3)$, $x' = (1, 0.1)$ and $x'' = (0.8, 0.9)$ giving the corresponding values $\lambda_1^* = 0$ and $\lambda_2^* = 0.25$. Hence, we have $l(\lambda_1^*) = \delta(x^*, (1, 0.1)) = (1, 0)$ and $l(\lambda_2^*) = \delta(x^*, (0.95, 0.3)) = (0, 1)$ and hence $d(l(\lambda_1^*), l(\lambda_2^*)) = 2$.

On the other hand $\lambda^*_{\pi(t)} < 1$ may be relaxed as shown in the next corollary.

Corollary 2.12. *Corollary 2.10 remains valid in the case*

$$\lambda^*_{\pi(t)} = max\{\lambda_j^* : j \in N^{\neq}\} = 1.$$

Proof. It suffices to prove that the chosen tie breaking rule (14) may be extended to the case $x^*_{\pi(t)} \in [0, 1]$. Since the equation $\lambda^*_{\pi(t)} = 1$ implies $x^*_{\pi(t)} = x''_{\pi(t)}$ then $\delta(x^*_{\pi(t)}, x^*_{\pi(t)})$ may be defined as

$$\delta(x^*_{\pi(t)}, x^*_{\pi(t)}) = \begin{cases} x''_{\pi(t)} & \text{if } x''_{\pi(t)} \in \{0, 1\} \\ 1 & \text{if } x''_{\pi(t)} > x'_{\pi(t)} \\ 0 & \text{if } x''_{\pi(t)} < x'_{\pi(t)} \end{cases} \tag{16}$$

However, the case $x''_{\pi(t)} \in \{0, 1\}$ is already contained in the other two cases (which represent the imposed tie breaking rule (14)). Indeed, if $x''_{\pi(t)}$ is equal to 1, then $x''_{\pi(t)} > x'_{\pi(t)}$ (since $x''_{\pi(t)} \neq x'_{\pi(t)}$), which corresponds to the second case, while if $x''_{\pi(t)}$ is equal to 0, then $x''_{\pi(t)} < x'_{\pi(t)}$, which corresponds to the last case. Consequently, the definition of $l_j(\lambda)$ given in Eq. (15) is valid in the case $\lambda^*_{\pi(t)} = 1$ as well as in the case given by Corollary 2.10. □

Another way to see the necessity of the condition $\lambda^*_{\pi(1)} = min\{\lambda_j^* : j \in N^{\neq}\} > 0$ is to note that it is not possible to extend the imposed tie breaking rule (14) to the case $x^*_{\pi(1)} \in [0, 1]$. Indeed, since $\lambda^*_{\pi(1)} = 0$ compels $x^*_{\pi(1)} = x'_{\pi(1)}$ then $\delta(x^*_{\pi(1)}, x^*_{\pi(1)})$ may be defined as

$$\delta(x^*_{\pi(1)}, x^*_{\pi(1)}) = \begin{cases} x'_{\pi(1)} & \text{if } x'_{\pi(1)} \in \{0, 1\} \\ 1 & \text{if } x''_{\pi(1)} > x'_{\pi(1)} \\ 0 & \text{if } x''_{\pi(1)} < x'_{\pi(1)} \end{cases} \tag{17}$$

Now, it is easy to check that the first case in Eq. (17) cannot be omitted while the definition of $\delta(x^*_{\pi(1)}, x^*_{\pi(1)})$ remains in force, i.e., the first case in Eq. (17) is not contained in the remaining two cases (the proof is similar to the proof of Corollary 2.12).

However, if we change the imposed tie breaking rule (14) such that

$$\delta(x_j^*, x_j^*) = \begin{cases} 1 & \text{if } x''_j < x'_j \\ 0 & \text{if } x''_j > x'_j \end{cases} \tag{18}$$

then it can be easily demonstrated that Corollary 2.10 will be true if $\lambda^*_{\pi(1)} = min\{\lambda_j^* : j \in N^{\neq}\} = 0$, but will not be true if $\lambda^*_{\pi(t)} = max\{\lambda_j^* : j \in N^{\neq}\} = 1$. In other words, informally speaking, complementing the tie breaking rule corresponds to complementing the necessary conditions.

It should be emphasized that if we change the definition of directional rounding so that $\delta(x_j^*, x'_j)$ is defined as

$$\delta(x_j^*, x'_j) = \begin{cases} 0 & \text{if } x'_j < x_j^* \\ 1 & \text{if } x'_j > x_j^* \\ 0 \text{ or } 1 & \text{otherwise} \end{cases} \tag{19}$$

and if we suppose that we use the tie breaking rule from Corollary 2.10 for the last case, then Corollary 2.10 will hold when $\lambda^*_{\pi(1)} = 0$ or $\lambda^*_{\pi(t)} = 1$ and even when $\lambda^*_{\pi(1)} < 0$ or $\lambda^*_{\pi(t)} > 1$. The justification for this claim is based on the fact that in this case the components $l_j(\lambda^*_{\pi(h)})$ of any vector $l(\lambda^*_{\pi(h)})$ will be given by Eq. (15).

In the next lemma we consider case where certain λ^* values are the same and show the relation among the corresponding $l(\lambda^*)$ vectors, thereby generalizing the preceding results.

Lemma 2.13. *Let x' and x'' be focal points and let x^* be a base point, where $\lambda_j^* = \frac{x_j^* - x'_j}{x''_j - x'_j}$ for $j \in N^{\neq}$. If $\lambda^*_{\pi(1)} < \lambda^*_{\pi(2)} < \cdots < \lambda^*_{\pi(h-1)} < \lambda^*_{\pi(h)} = \lambda^*_{\pi(h+1)} = \cdots = \lambda^*_{\pi(h+k)} < \cdots < \lambda^*_{\pi(t)}$, $k \geq 1$ then*

$$l_j(\lambda^*_{\pi(h-1)}) = l_j(\lambda^*_{\pi(h)}) = \cdots = l_j(\lambda^*_{\pi(h+k)}) \text{for } j \in N - \{\pi(h-1), \pi(h), \ldots, \pi(h+k)\}.$$

Moreover, if $\lambda^*_{\pi(1)} > 0$ and $\lambda^*_{\pi(t)} < 1$ and ties in directional rounding $\delta(x_j^*, x_j^*)$ which arise in the case $0 < x_j^* < 1$ are broken by setting

$$\delta(x_j^*, x_j^*) = \begin{cases} 1 & \text{if } x''_j > x'_j \\ 0 & \text{if } x''_j < x'_j \end{cases} \tag{20}$$

then

$$l_{\pi(h+p)}(\lambda^*_{\pi(h)}) = 1 - l_{\pi(h+p)}(\lambda^*_{\pi(h-1)}) \text{ for } p = 0, 1, \ldots, k$$

Proof. First, note that vectors $l(\lambda^*_{\pi(h)}), l(\lambda^*_{\pi(h+1)}), \ldots, l(\lambda^*_{\pi(h+k)})$ are the same since $\lambda^*_{\pi(h)} = \lambda^*_{\pi(h+1)} = \cdots = \lambda^*_{\pi(h+k)}$. If $x'_j = x''_j$ and $j \in N - \{\pi(h-1), \pi(h), \ldots, \pi(h+k)\}$ the conclusion of the lemma is immediate. On the other hand if $x'_j \neq x''_j$ and $\lambda^*_{\pi(j)} < \lambda^*_{\pi(h-1)} < \cdots < \lambda^*_{\pi(h+k)}$ or $\lambda^*_{\pi(j)} > \lambda^*_{\pi(h+k)} > \cdots > \lambda^*_{\pi(h)}$ then the definition of the vectors $l(\lambda)$ implies that $l_j(\lambda^*_{\pi(h-1)}) = l_j(\lambda^*_{\pi(h)}) = \cdots = l_j(\lambda^*_{\pi(h+1)}) = l_j(\lambda^*_{\pi(h+k)})$. However, by Corollary 2.10 which establishes that $l_{\pi(j+1)}(\lambda^*_{\pi(j+1)}) = 1 - l_{\pi(j)}(\lambda^*_{\pi(j)})$ for each pair $(\pi(h+p), \pi(h-1))$, for $p = 0, 1, \ldots, k$ we obtain

$$l_{\pi(h+p)}(\lambda^*_{\pi(h+p)}) = 1 - l_{\pi(h+p)}(\lambda^*_{\pi(h-1)}) \text{ for } p = 0, 1, \ldots, k$$

which is equivalent to

$$l_{\pi(h+p)}(\lambda^*_{\pi(h)}) = 1 - l_{\pi(h+p)}(\lambda^*_{\pi(h-1)}) \text{ for } p = 0, 1, \ldots, k. \quad \square$$

In the same way as in Corollary 2.10 it can be shown that if $0 < \lambda^*_{\pi(1)} < \lambda^*_{\pi(2)}, \ldots, \lambda^*_{\pi(h-1)} < \lambda^*_{\pi(h)} = \lambda^*_{\pi(h+1)} = \cdots = \lambda^*_{\pi(h+k)} < \ldots, < \lambda^*_{\pi(t)} \leq 1$ and if ties are broken as proposed in Corollary 2.10 (and Lemma 2.13) then

$$l_j(\lambda^*_{\pi(h+k+1)}) = l_j(\lambda^*_{\pi(h)}) \text{ for } j \in N - \{\pi(h+k+1)\}$$

$$l_{\pi(h+k+1)}(\lambda^*_{\pi(h+k+1)}) = 1 - l_{\pi(h+k+1)}(\lambda^*_{\pi(h)}).$$

In order to produce a larger number of solutions and therefore to increase the chance to reach an optimal solution we can treat $\lambda^*_{\pi(h+p)}$ for $p = 0, 1, \ldots, k$ as distinct values and associate to each of them the vectors defined as

$$l_j(\lambda^*_{\pi(h+p)}) = \begin{cases} l_j(\lambda^*_{\pi(h+p-1)}) & \text{if } j \neq \pi(h+p) \\ 1 - l_j(\lambda^*_{\pi(h+p-1)}) & \text{if } j = \pi(h+p) \end{cases} \tag{21}$$

The vector $l(\lambda^*_{\pi(h+k)})$ then corresponds to the vector $l(\lambda^*_{\pi(h)})$ defined in the previous lemma.

2.4. Exploiting the results to generate Star Paths

To generate a Star Path $L(x^*, x', x'')$ where $Line(x', x'')$ is restricted to the segment $[x', x'']$ we propose the following Algorithm 1 which exploits the previously proven statements. Since we consider just the segment $[x', x'']$, we want to produce a set of vectors obtained by directional rounding using points which belong to the segment as focal points and x^* as a base point, i.e., $L(x^*, x', x'') = \{l(\lambda) = \delta(x^*, x) : x = x' + \lambda(x'' - x'),\ 0 \leq \lambda \leq 1\}$. According to Lemma 2.5 the Star Path $L(x^*, x', x'')$ can be constructed by rounding directionally only a finite number of points from the segment $[x', x'']$, i.e., $L(x^*, x', x'') = \{l(\lambda) : \lambda \in \{0, 1\} \text{ or } \lambda = \lambda^*_j, \text{ with } 0 \leq \lambda^*_j \leq 1,\ j \in N^{\neq}\}$. Further as shown in the remaining statements there are benefits caused by sorting λ^*_j, so we assume that all λ^*_j with values between 0 and 1 are sorted in non-decreasing order, i.e., $0 \leq \lambda^*_{\pi(1)} \leq \lambda^*_{\pi(2)} \leq \cdots \leq \lambda^*_{\pi(p)} \leq 1$. In the case that $\lambda^*_{\pi(1)} > 0$, we can add an artificial $\lambda^*_{\pi(0)}$ with value 0 in order to cover case $\lambda = 0$ (note in this case $\pi(0)$ does not correspond to any index).

On the other hand if $\lambda^*_{\pi(p)} < 1$ there is no need to extend the array of λ^*_j values since according to Lemma 2.5 $l(1) = l(\lambda^*_{\pi(p)})$. Hence, without loss of generality we may suppose that $\lambda^*_{\pi(0)} = 0$. Based on these observations, the generation of the Star Path $L(x^*, x', x'')$ may be constructed in the following way. The first vector to be generated is the vector $l(\lambda^*_{\pi(0)})$ which is obtained using the definition given in Lemma 2.5 together with the tie breaking rule.

From then on, each vector is derived from the immediately preceding vector using Corollary 2.10, Corollary 2.12 and Lemma 2.13 in the following manner. Let $N' = \{j \in N^{\neq} : 0 \leq \lambda^*_j \leq 1\}$, $n' = |N'|$ and let $N^0 = \{j \in N^{\neq} : \lambda^*_j = 0\}$, $n^0 = |N^0|$. Then the vector $l(\lambda^*_{\pi(n^0+1)})$ may be obtained from the vector $l(\lambda^*_{\pi(0)})$ by adjusting values at the positions $\pi(1), \ldots, \pi(n^0)$ according to the rules for generating vector $l(\lambda^*_{\pi(n^0+1)})$. In order to avoid (possible) large distances between vectors $l(\lambda^*_{\pi(0)})$ and $l(\lambda^*_{\pi(n^0+1)})$ we may generate additional vectors $l(\lambda^*_{\pi(i)})$, $i = 1, \ldots, n'$ obtained from the previous vector by adjusting entries at the position $\pi(i)$ according to the rules for generating vector $l(\lambda^*_{\pi(n^0+1)})$. Then each element $l(\lambda^*_{\pi(h)})$, $h > n'$ can be derived directly from the vector $l(\lambda^*_{\pi(h-1)})$, by just complementing the value at the position $\pi(h)$ (Corollary 2.10, Corollary 2.12 and Lemma 2.13). Note that all $\lambda^*_{\pi(h)}$, $h > n^0$, are treated as distinct in order to increase the number of generated vectors as explained above.

Algorithm 1 Generating Star Path $L(x^*, x', x'')$

Function Star Path(x^*, x', x'')

Compute $\lambda_j^* = \frac{x_j^* - x_j'}{x_j'' - x_j'}$ for each $j \in N^{\neq} = \{j \in N : x_j' \neq x_j''\}$;
Let $N' = \{j \in N^{\neq} : 0 \leq \lambda_j^* \leq 1\}$, and let $N^0 = \{j \in N^{\neq} : \lambda_j^* = 0\}$, set $n' = |N'|$ and $n^0 = |N^0|$;
Sort all λ_j^* for $j \in N'$ in nondecreasing order, i.e., so that $\lambda_{\pi(1)}^* \leq \lambda_{\pi(2)}^* \leq \cdots \leq \lambda_{\pi(n')}^*$;
Set $\lambda_{\pi(0)}^* = 0$;
Determine $x = l(\lambda_{\pi(0)}^*) = \delta(x^*, x')$ using Lemma 2.5;
Set $L = \{x\}$;
for $j = 1$ to n^0 **do**
  $x_{\pi(j)} = \ell_{\pi(j)}(\lambda_{\pi(n^0+1)}^*)$;
  $L = L \cup \{x\}$;
end for
for $j = n^0 + 1$ to n' **do**
  $x_{\pi(j)} = 1 - x_{\pi(j)}$;
  $L = L \cup \{x\}$;
end for
return L;

In what follows we show that the directional rounding of some sets may be performed in an efficient way, too. We first show that the directional rounding of a certain cone may be performed by directionally rounding just one line segment lying within it. This property stems from the following theorem which identifies line segments having a common endpoint, whose directional roundings produce the same set of 0–1 points.

Theorem 2.14. *Given x' and x'' as focal points and x^* as a base point, $\delta(x^*, [x', x'']) = \delta(x^*, [y, x''])$ where $y \in Ray(x^*, x') - \{x^*\}$. Furthermore, $\delta(x^*, [x', x'']) = \delta(x^*, [x', z])$ where $z \in Ray(x^*, x'') - \{x^*\}$.*

Proof. Let y be any point on the ray from x^* through x' different from x^*. Hence vector y will be represented as $y = x^* + \alpha(x' - x^*)$ with $\alpha > 0$. According to Lemma 2.4, we have $\delta(x^*, x') = \delta(x^*, y)$. Consequently, to show that $\delta(x^*, [x', x'']) = \delta(x^*, [y, x''])$ it suffices to show that the value of $\lambda_j^* = \frac{x_j^* - x_j'}{x_j'' - x_j'}$ values sorted in nondecreasing order retains this order by replacing x_j' with y_j. Indeed, from Algorithm 1 it follows that each Star Path is uniquely determined by the starting point of line segment and the order of λ_j^* values. Let $\lambda'^* = \frac{x_j^* - y_j}{x_j'' - y_j}$ and suppose we have the order $0 \leq \lambda_1^* \leq \lambda_2^* \leq \cdots \leq \lambda_{n'}^* \leq 1$. Then, for each j, $\lambda_j'^*$ may be written as $\lambda_j'^* = \frac{\alpha(x_j^* - x_j')}{x_j'' - x_j^* + \alpha(x_j^* - x_j')}$ or equivalently as $\lambda_j'^* = \frac{\alpha(x_j^* - x_j')}{x_j'' - x_j' + (\alpha - 1)(x_j^* - x_j')}$. Now, from the definition of the λ_j^* values and the last expression of $\lambda_j'^*$ we have $\lambda_j'^* = \frac{\alpha\lambda_j^*}{1 - \lambda_j^* + \alpha\lambda_j^*}$. Next, it is easy to check that $\lambda_j'^* \leq \lambda_k'^* \Leftrightarrow \lambda_j^* \leq \lambda_k^*$. Hence,

$$0 \leq \lambda_1^* \leq \lambda_2^* \leq \cdots \leq \lambda_{n'}^* \leq 1 \Leftrightarrow 0 \leq \lambda_1'^* \leq \lambda_2'^* \leq \cdots \leq \lambda_{n'}'^* \leq 1$$

and therefore $\delta(x^*, [x', x'']) = \delta(x^*, [y, x''])$.

In a similar way, if we choose any point $z \neq x^*$ on the ray from x^* through x', i.e., $z = x^* + \alpha(x'' - x^*)$ with $\alpha > 0$, it holds $\delta(x^*, x'') = \delta(x^*, z)$ (according to Lemma 2.4). Further if $\lambda''^* = \frac{x_j^* - z_j}{x_j'' - z_j}$, we obtain, analogously to the previous case, $\lambda_j''^* = \frac{\lambda_j}{\lambda_j + \alpha}$ for each j. Then it follows that $\lambda_j''^* \leq \lambda_k''^* \Leftrightarrow \lambda_j^* \leq \lambda_k^*$ and therefore $\delta(x^*, [x', x'']) = \delta(x^*, [x', z])$. □

As a consequence of Theorem 2.14, we have the following result.

Corollary 2.15. *Given x' and x'' as focal points and x^* as a base point, $\delta(x^*, [x', x'']) = \delta(x^*, [y, z])$ where $y \in Ray(x^*, x') - \{x^*\}$ and $z \in Ray(x^*, x'') - \{x^*\}$.*

This corollary effectively says that $\delta(x^*, [x', x'']) = \delta(x^*, Cone(x^*, \{x', x''\}) - \{x^*\})$.

We finish this section by showing that sometimes to directionally round a certain convex set it suffices to directionally round just one of its extreme points.

Theorem 2.16. *Given x^* as a base point and X as a set of focal points, if $\delta(x^*, X) = x'$ then $\delta(x^*, conv(X)) = x'$.*

Proof. If $y \in conv(X)$ then there exists a set of points $x^1, \ldots, x^p$ such that $y = \sum_{i=1}^p \alpha_i x^i$ and $\sum_{i=1}^p \alpha_i = 1$ with $\alpha_i \geq 0$ and $x^i \in X$ for $i = 1, \ldots, p$. It is easy to see that we have $y_j \geq x_j^*, j \in N$ if $x_j^i \geq x_j^*$ for all $i = 1, \ldots, p$ and $y_j \leq x_j^*$ if $x_j^i \leq x_j^*$ for all $i = 1, \ldots, p$. Hence, from the definition of directional rounding δ, if we assume that "tie breaking" rule is the same for all solutions x^i $(i = 1, \ldots, p)$ when $x^i_j = x^*_j$ with $0 < x^*_j < 1$ for all $j \in N$ we have 03 $\delta(x^*_j, y_j) = \delta(x^*_j, x^i_j)$. □

3. Fundamental analysis of Star Paths with directional rounding

In this section, we review some basic properties of 0–1 Mixed Integer Programs and provide some basic results that motivate the use of directional rounding in solving 0–1 Mixed Integer Programs. Before this we review the bounded simplex method, an efficient method to solve the LP relaxation of the MIP problem.

3.1. Standard LP basic solution representation

The bounded simplex method proposed by Dantzig [12] is an efficient method to solve the LP relaxation of the MIP problem by systematically exploring extreme points of solution space. The search for an optimal extreme point is performed by pivot operations, each of which moves from one extreme point to an adjacent extreme point by removing one variable from the current basis and bringing another variable (which is not in the current basis) into the basis. For our purposes, the procedure can be depicted in the following way. Suppose that the method is currently at some extreme point $x(0)$ with corresponding basis B. The set of indices of all other variables (nonbasic variables) will be designated with $\overline{B} = N - B$. The extreme points adjacent to $x(0)$ have the form

$$x(j) = x(0) - \theta_j D_j \text{ for } j \in \overline{B} \tag{22}$$

where D_j is a vector associated with the nonbasic variable x_j, and θ_j is the change in the value of x_j that moves the current solution from $x(0)$ to $x(j)$ along their connecting edge. The *LP* basis representation identifies the components D_{kj} of D_j, as follows

$$D_{kj} = \begin{cases} ((A^B)^{-1}A)_{kj} & \text{if } k \in B \\ \xi \quad \text{if } k = j \\ 0 \quad \text{if } k \in \overline{B} - \{j\} \end{cases} \tag{23}$$

where A^B represents the matrix obtained from matrix A by selecting columns that correspond to the basic variables and $\xi \in \{-1, 1\}$. We choose the sign convention for entries of D_j that yields a coefficient for x_j of $D_{jj} = 1$ if x_j is currently at its lower bound at the vertex $x(0)$, and of $D_{jj} = -1$ if x_j is currently at its upper bound at $x(0)$. We assume throughout the following that $x(0)$ is feasible for the LP problem, though this assumption can be relaxed. In general, we require that $x(0)$ is a basic solution (feasible or not) — i.e., $x(0)$ is a feasible extreme point for a region that results by discarding some of the constraints that define the original LP feasible region.

Note that if we consider an extreme point $x(0)$ and its adjacent extreme points $x(j)$ for $j \in \overline{B}$, we can conclude that the points $x(j)$ for $j \in \overline{B}$ are linearly independent and that point $x(0)$ does not belong to the plane spanned by these points. Furthermore, this observation holds even when these θ_j values are replaced by any positive value.

3.2. Justification of Star Paths with directional rounding

We start this section by presenting some properties of the LP polyhedron of a 0–1 MIP (see, for example, Dantzig [12]).

Lemma 3.1. *Let $x(0)$ be a basic extreme point associated with a basis B, and define $x(j)$ for $j \in \overline{B}$ by (22) for any given positive value θ^*_j for θ_j, i.e.,*

$$x(j) = x(0) - \theta^*_j D_j, j \in \overline{B} \tag{24}$$

Then, we obtain:

$$Cone(x(0), X(\overline{B})) = \{x(0) - \sum_{j \in \overline{B}} \lambda_j D_j : \lambda_j \geq 0, j \in \overline{B}\}. \tag{25}$$

Proof. The proof follows directly from the definition of the cone. □

Lemma 3.2. *Let $x(0)$ be a basic extreme point with its associated basis denoted by B. Then all feasible solutions of the LP problem belong to the $Cone(x(0), X(\overline{B}))$.*

Proof. Without loss of generality we can assume that $B = \{1, 2, \ldots m\}$, where m represents the number of rows in the matrix A and let $\mathcal{B} = (A^B)^{-1}$. Therefore, each solution x may be represented as $x = [x_B, x_{\overline{B}}]^T$. Furthermore, each LP-feasible solution x can be expressed as $x_B = \mathcal{B}b - \mathcal{B}A^{\overline{B}}x_{\overline{B}}$. This last equality may be rewritten as $x_B = \mathcal{B}b - [\mathcal{B}A^{m+1}, \ldots \mathcal{B}A^n]x_{\overline{B}}$. Defining a new set of variables $\theta_{\overline{B}} = x(0)_{\overline{B}} - x_{\overline{B}}$ the last equation becomes $x_B = \mathcal{B}b - [\mathcal{B}A^{m+1}, \ldots \mathcal{B}A^n]x(0)_{\overline{B}} +$

$[\mathcal{B}A^{m+1},\dots\mathcal{B}A^{n}]\theta_{\overline{B}}$ or equivalently $x_B = x(0)_B + [\mathcal{B}A^{m+1},\dots\mathcal{B}A^{n}]\theta_{\overline{B}}$. Using the last equation we obtain the following representation of the solution x

$$x = x(0) - \begin{bmatrix} \mathcal{B}A^{m+1} & \mathcal{B}A^{m+2} & \cdots & \mathcal{B}A^{n} \\ \xi & 0 & \cdots & 0 \\ 0 & \xi & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi \end{bmatrix} \theta_{\overline{B}} \tag{26}$$

However adjusting entries of matrix in (26) according to replacing $\theta_{\overline{B}}$ with $|\theta_{\overline{B}}|$ yields a matrix whose columns are directional vectors $D_j, j \in \overline{B}$, and therefore $x = x(0) - \sum_{j\in\overline{B}} D_j|\theta_j|$, i.e., $x \in Cone(x(0), X(\overline{B}))$. Since x is an arbitrarily chosen feasible point we conclude that all feasible points belong to the $Cone(x(0), X(\overline{B}))$. □

Corollary 3.3. *All IP feasible solutions belong to the* $Cone(x(0), X(\overline{B}))$.

Lemma 3.4. *Let* $x(0)$ *be an LP feasible extreme point with associated basis B, then* $x(0)$ *does not belong to* $Half\_Space(X(\overline{B}))$*, but belongs to the complementary half space.*

Proof. By definition, we have $Plane(X(\overline{B})) = \{x(0) - \sum_{j\in\overline{B}} \lambda_j D_j : \sum_{j\in\overline{B}} \lambda_j = 1\}$. Hence, its corresponding half space is $Half\_Space(X(\overline{B})) = \{x(0) - \sum_{j\in\overline{B}} \lambda_j D_j : \sum_{j\in\overline{B}} \lambda_j \geq 1\}$. Keeping in mind the linear independence of vectors $D_j, j \in \overline{B}$ it follows that $Half\_Space(X(\overline{B}))$ does not contain point $x(0)$. Consequently, $x(0)$ belongs to the complementary half space. □

Lemma 3.5. *Let* $x(0)$ *be an LP feasible extreme point with the associated basis B. Then all optimal IP solutions, excluding* $x(0)$*, belong to* $Half\_Space(X(\overline{B}))$.

Proof. The lemma follows directly from the fact that $Half\_Space(X(\overline{B}))$ may be considered as a valid cutting plane that excludes $x(0)$ as feasible. □

Corollary 3.6. *The previous two lemmas remain true if the set* $X(\overline{B})$ *is replaced by the set* $\{x(0) - \theta_j^* D_j : j \in \overline{B},\ 0 < \theta_j^* < \theta_j\}$.

The next two theorems represent the fundamental results that motivate the use of directional rounding to solve 0–1 MIP.

Theorem 3.7. *Let* $x(0)$ *be an LP feasible extreme solution with associated basis B, and let* $X(\overline{B}) = \{x(0) - \theta_j^* D_j : j \in \overline{B}\}$ *where* θ_j^* *is any given positive value. Then for any IP feasible solution* x' *there is a convex region* $X \subset Face(X(\overline{B}))$ *such that* $\delta(x(0), x) = x'$ *for all* $x \in X$*. Moreover, if X is not polyhedral, there is a polyhedral subset of X for which this conclusion is true.*

Proof. Given that all IP feasible solutions are in the $Cone(x(0), X(\overline{B})) = \{x(0) - \sum_{j\in\overline{B}} \lambda_j D_j : \lambda_j \geq 0\}$, it follows that for each IP feasible solution x' the $Ray(x(0), x')$ which belongs to that cone intersects the $Face(X(\overline{B}))$. Denote this intersection point by y, hence $y = Ray(x(0), x') \cap Face(X(\overline{B}))$. According to the previous lemmas the point y satisfies $\delta(x(0), y) = \delta(x(0), x') = x'$. The last condition means that for any IP feasible solution x', there is at least one solution which belongs to the $Face(X(\overline{B}))$ and which produces the solution x' by directional rounding relative to the base solution $x(0)$. However, if there exists more than one solution which may be directionally rounded to the yield x', then according to Theorem 2.16 the directional rounding of any solution in the convex hull X of these points, produces the same solution x'. Further, if X is not polyhedral, its polyhedral subset can be identified as a set of all convex combinations of a finite number of points from X. □

Corollary 3.8. *The previous theorem is also valid when the word "feasible" is replaced by "optimal".*

The next result is well known [46], but we include it to facilitate the proof of Theorem 3.10.

Lemma 3.9. *An optimal solution for the 0–1 IP problem may be found at an extreme point of the LP feasible set.*

Proof. Since all continuous variables are slack variables in the IP problem considered in this paper, the IP feasible set can be viewed as $X = \{x : Ax \leq b,\ 0 \leq x_j \leq 1,\ j \in I\ x \in \mathbb{R}^{|I|}\}$. In other words, set X is intersection of a convex set $\{x : Ax \leq b,\ x \in \mathbb{R}^{|I|}\}$ and a cube $\{x : 0 \leq x_j \leq 1,\ \ j \in I\ x \in \mathbb{R}^{|I|}\}$. Therefore, since all 0–1 solutions are located at the extreme points of the unit cube, all feasible 0–1 solutions are also located at the extreme points of X. □

Theorem 3.10. *Let* $x(0)$ *be an LP optimal extreme point and let C be a valid cutting plane for problem IP, that excludes* $x(0)$ *and is satisfied by all IP feasible solutions. Then, if problem IP has an optimal solution there exists a polyhedral region* $P \subset C$ *such that each IP feasible solution can be obtained by directional rounding of some point in P. Moreover, at least one optimal IP solution is obtained by directional rounding of some extreme point of P.*

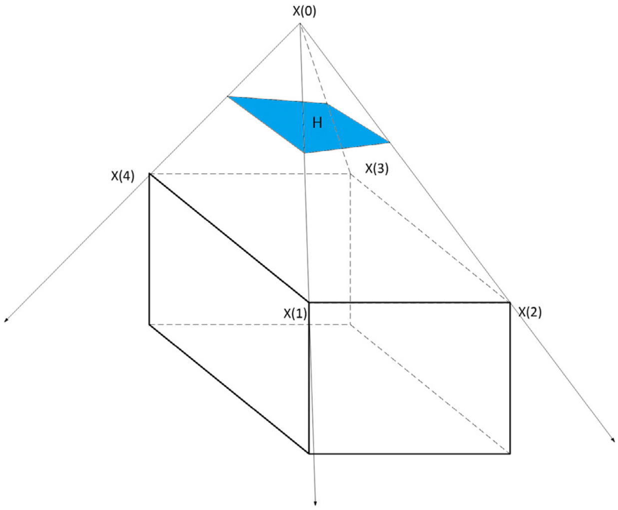

Fig. 2. Cone $C(x(0), X(\bar{B}))$ and valid cutting plane C.

Proof. Since cutting plane C separates $x(0)$ from all IP feasible solutions, each ray starting at $x(0)$ through some IP feasible solution, say x, intersects the plane C at some point, say x', i.e., $x' = Ray(x(0), x) \cap C$. However, Lemma 2.4 demonstrates that directional rounding of x' gives the point x. Therefore each IP feasible solution can be obtained by rounding some point from the region $P = conv(\{Ray(x(0), x) \cap C : x \text{ } IP \text{ } feasible \text{ } solution\})$. This set P is a polyhedral region since it is a convex hull of a finite number of IP feasible solutions. Hence, P may be rewritten by reference to the convex hull of its extreme points $x'^1, x'^2, \ldots, x'^p$ as $P = conv(\{x'^i : i = 1, \ldots, p\})$. If X is the IP feasible set, then $X \subseteq \delta(x(0), P)$.

Assume that none of the solutions $x^i = \delta(x(0), x'^i)$ for $i = 1, \ldots, p$, is IP optimal. Therefore, using Lemma 3.9, we can prove that $x(0)$ and an optimal solution for x^{opt} IP are on different sides of plane $H = \{\sum_{i=1}^{p} \lambda_i x^i : \sum_{i=1}^{p} \lambda_i = 1\}$ (see Fig. 2). (If we suppose $x(0)$ and x^{opt} are on the same side then x^{opt} is inside the truncated cone, defined by rays from $x(0)$ through points x'^i and H, contradicting the fact that x^{opt} is an extreme point.) Hence, there is a point $y \in H$ such that $y = x(0) + \lambda(x^{opt} - x(0))$, $0 < \lambda < 1$. Using the linearity of the objective function and the optimality of the LP-solution $x(0)$ we conclude that $cx(0) \leq cy \leq cx^{opt}$. On the other hand, since y belongs to H it can be represented as $\sum_{i=1}^{p} \lambda_i x^i$, $\sum_{i=1}^{p} \lambda_i = 1$, so $cy = \sum_{i=1}^{p} \lambda_i c x^i > cx^{opt}$ because none of the points x^i is IP optimal. This contradicts our starting assumption that none of the points x^i is IP optimal. Hence, at least one solution among solutions $x^i = \delta(x(0), x'^i)$ for $i = 1, \ldots, p$ is an optimal IP solution and consequently can be obtained by directional rounding of some extreme point of P. □

Corollary 3.11. *The previous lemmas and theorems hold when $x(0)$ is any basic solution (feasible or not) with associated basis B.*

4. Convergent Scatter Search with directional rounding

We now build on the fact that Scatter Search consists of a systematic exploration of solution space relative to a set of reference points. As pointed out, these points typically consist of good solutions obtained by prior problem solving effort, where the term "good solution" refers not only to solutions with good objective function values, but also to solutions which increase the diversity of the set of reference points.

4.1. Variant of convergent scatter search

In this section we propose a version of scatter search with directional rounding that converges in a finite number of iterations to an optimal solution for the 0–1 MIP problem. The resulting Convergent Scatter Search algorithm is justified by the preceding theorems. Let $x(0)$ be an optimal solution of the LP relaxation with the associated basis B and let $\pi x \leq \pi_0$ be a valid cutting plane for problem IP that excludes $x(0)$ without excluding any IP feasible solutions. Valid cutting planes are very well studied in the literature. There are several procedures to generate deep cuts including convexity cuts or intersection cuts (see for example, [3,4,7,17,44,47]). Recently, Balas and Margot [6] introduced a generalization of the intersection cuts that dominate the original ones. We define the polyhedral region P as the intersection of the hyperplane $\pi x = \pi_0$ with the set of feasible solutions of the LP relaxation $\overline{X}$, i.e., $P = \{x : \pi x = \pi_0\} \cap \overline{X}$. By the preceding results we know that an optimal IP solution belongs to the set $\delta(x(0), P)$. The Convergent Scatter Search algorithm works as follows.

At each iteration, we choose an extreme point of the current polyhedron (polyhedral cone) P, say x' and select a cone $C(x')$ originating at x' such that it is easy to compute the finite set $\delta(x(0), C(x'))$. The process is repeated on the new polyhedron $P = P - C(x')$ until the volume of the current polyhedron P is null. Thus, the algorithm successively nibbles away portions $C(x')$ starting from the corner x' each time until nothing is left (see Algorithm 2). In each iteration, the volume of the polyhedron P may be computed as explained in [35], for example. This value identifies the portion of polyhedron not yet explored. However, the computation of the volume may be avoided during the execution of the algorithm until P becomes the empty set.

Algorithm 2 Framework of the Convergent Scatter Search algorithm $CSS()$

Function `CSS()`

Let $x(0)$ be an optimal solution of LP relaxation;
Set $P = \{x : \pi x = \pi_0\} \cap \overline{X}$, where $\pi x \leq \pi_0$ is a cutting plane;
while $Volume(P) > 0$ **do**
  Choose any extreme point of P, i.e., x';
  Construct the cone $C(x')$ originating at x';
  Compute $S = \delta(x(0), C(x'))$;
  Update the best solution $x^* = \arg\min_{x \in S \cup x^*} cx$;
  Reduce the current polyhedral $P = P - C(x')$;
end while
return x^*;

Now, we explain how we construct at each iteration the cone $C(x')$ originating at an extreme point x' of the current polyhedron P. Let B denote the basis associated with the extreme point x' relative to the polyhedron P. For each adjacent point of x', $x'(h)$, $h \in \overline{B}$ we determine $y'(h)$ belonging to the $Ray(x', x'(h))$ such that directional rounding of all points from open sub-edge $]x', y'(h)[$ yields the same 0–1 solution. These points $y'(h)$ can be computed using Theorem 2.6 or the first iteration of Algorithm 1. More precisely, $y'(h) = x' + \lambda^*(x'(h) - x')$ with $\lambda^* = min\{\lambda_j^* : 0 < \lambda_j^* = \frac{x(0)_j - x'_j}{x'(h)_j - x'_j} \leq 1\}$. The constructed cone originating at the extreme point x' is defined as $C(x') = Cone(x', Y) \cap H^-$ where $Y = \{y'(h) : h \in \overline{B}\}$ and $H^- = \{\sum_{h \in \overline{B}} \lambda_h y'(h) : \sum_{h \in \overline{B}} \lambda_h < 1\}$.

A Convergent Scatter Search algorithm may be implemented in two different ways according to whether or not we reduce the size of the LP polyhedron at the same time that we reduce the size of $Plane(X(\overline{B}))$. These two different algorithmic approaches are depicted as Algorithms 3 and 4 respectively. Both start by solving the LP relaxation and creating $Plane(X(\overline{B}))$ as a convex hull of all extreme points adjacent to an optimal solution of the LP relaxation. Then both methods choose the best extreme point x' (the point that minimizes the objective function of the initial problem) which belongs to the set $X(\overline{B})$ and determine the cone $C(x')$ to be directionally rounded. Then the truncated cone $C(x')$ is directionally rounded using Theorem 4.2. The first algorithm (Algorithm 3) then continues to repeat the overall process on $Plane(X(\overline{B})) - C(x')$ and finishes its work when the difference between the objective function values at x' and the best encountered integer solution is less than a fixed ϵ. On the other hand, Algorithm 4 repeats the overall process on the new MIP problem derived from the initial problem by imposing a valid cut which excludes the optimal solution of LP relaxation and the pseudo cut which eliminates extreme point x'. The pseudo cut is determined as a hyperplane passing through points from the set Y and the optimal solution of the LP relaxation (which are linearly independent). The second algorithm stops when the difference between the value of the LP relaxation of the new created problem and the value of best integer solution found is less than a fixed ϵ. (For a pure IP problem if all the data are integers the value of ϵ can be set to 1.) The coefficients of the hyperplane containing $Plane(X(\overline{B}))$, or the coefficients of the hyperplane passing through points from the set Y and $x(0)$, may be easily determined as a feasible non-zero solution of a linear program with a trivial objective function subject to $\pi x = \pi_0$ for $x \in \overline{B}$ or $x \in Y \cup \{x(0)\}$.

Algorithm 3 Convergent Scatter Search algorithm -version 1 $CSS1()$

Function `CSS1()`

Let $x(0)$ be an optimal solution of LP relaxation;
Set $P = \{x : \pi x = \pi_0\} \cap \overline{X}$, where $\pi x \leq \pi_0$ is a cutting plane;
repeat
  Let x' be an optimal solution of $min\{cx : x \in P\}$;
  Construct the cone $C(x')$ originating at x';
  Compute $S = \delta(x(0), C(x'))$;
  Update the best solution $x^* \leftarrow \arg\min_{x \in S \cup x^*} cx$;
  Reduce the current polyhedral $P = P - C(x')$;
until $cx^* - cx' \geq \epsilon$
return x^*;

Algorithm 4 Convergent Scatter Search algorithm -version 2 *CSS*2()

Function `CSS2()`

repeat
  Let $x(0)$ be an optimal solution of LP relaxation, i.e., $min\{cx : x \in \overline{X}\}$;
  Set $P = \{x : \pi x = \pi_0\} \cap \overline{X}$, where $\pi x \leq \pi_0$ is a cutting plane;
  Let x' be an optimal solution of $min\{cx : x \in P\}$;
  Construct the cone $C(x')$ originating at x';
  Compute $S = \delta(x(0), C(x'))$;
  Update the best solution $x^* \leftarrow \arg\min_{x \in S \cup x^*} cx$;
  Compute hyperplane $H = \{x : hx = h_0\}$ as the one passing through points from the set Y and $x(0)$;
  Define H^+ as the half space bounded with H that does not contain x';
  Reduce the current LP polyhedral $\overline{X} = \overline{X} \cap P \cap H^+$;
until $cx^* - cx(0) \geq \epsilon$
return x^*;

Implementation of the Convergent Scatter Search procedure requires a tool for enumerating all adjacent extreme points of a given extreme point of a polyhedron. In some cases, if there is no degeneracy, enumeration of adjacent extreme points can be performed easily using the formula of (22). Unfortunately, even if there is no degeneracy at the starting polyhedron of the LP relaxation degeneracy will appear during execution of the Convergent Scatter Search procedure since at each iteration we add some cutting plane to the starting LP problem. In consequence, we are unable to enumerate all extreme points using just formula (22). Additionally, the Convergent Scatter Search procedure requires a lot of memory for rounding cone $C(x')$, making it unsuitable for large scale problems. Because the 0–1 MIP problem is NP hard, we cannot offer a polynomial bound on the running time of our approach. However, from a practical standpoint we introduce a collection of fast approximation methods in a sequel paper [43] that illustrate some of the possibilities for exploiting the theoretical results described here. In particular, we propose several One-Pass (non-iterated) heuristics based on Scatter Search and directional rounding that may be used for large scale problems. The versions of the methods tested are "first stage" implementations to establish the power of these methods in a simplified form. The aim of [43] is to demonstrate the efficiency of these first stage methods, which makes them attractive for use in situations where very high quality solutions are sought with an efficient investment of computational effort.

4.2. Convergence proof of scatter search

To the best of our knowledge, up to now convergence results have been obtained for only a few metaheuristics [10,22,27]. In this subsection, we provide background and the convergence proof of the Scatter Search and Star Paths with Directional Rounding algorithms described in the previous subsection. More precisely, we provide theorems that demonstrate how to construct the truncated cone $C(x')$ originating at x' and how to directionally round it.

Theorem 4.1. *Let H be any hyperplane such that $H \cap C(x') \neq \emptyset$ and $H \cap conv(Y) = \emptyset$. Then $\delta(x(0), C(x')) = \delta(x(0), H \cap C(x'))$.*

Proof. By the definition of directional rounding, the fact that the directional rounding of multiple points relative to the base point $x(0)$ can yield the same 0–1 solution implies that those points belong to the same sub-space bounded by the hyperplanes $x_j = x(0)_j$. For example, in two dimensions if the directional rounding of some point A is equal to point (1,1) (i.e., $\delta(x(0), A) = (1, 1)$), then A belongs to the sub-space $x_1 \geq x(0)_1,\ x_2 \geq x(0)_2$. In turn, this observation implies that in the case of directionally rounding a line (rather than a single point), the points that lie on two consecutive intersections of the line and a hyperplane $x_j = x(0)_j$ will be rounded directionally to the same 0–1 point. It is easy to check that the set $\{H \cap C(x') : H \text{ a hyperplane},\ H \cap C(x') \neq \emptyset,\ H \cap conv(Y) = \emptyset\}$ equals $C(x')$. Hence, to prove the theorem it is enough to show that each set $H \cap C(x')$ is intersected by the same hyperplanes $x_j = x(0)_j$. However, no line segment $]x', y'(h)[$ is cut by a hyperplane $x_j = x(0)_j$ (otherwise, there will be at least two points on some segment $]x', y'(h)[$ which yield two different 0–1 solutions by directional rounding). If some set $H \cap C(x')$ is intersected by some hyperplane $x_l = x(0)_l$ then this hyperplane passes through point x' and therefore hyperplane $x_l = x(0)_l$ must intersect all sets $H \cap C(x')$ such that H is a hyperplane, $H \cap C(x') \neq \emptyset$, $H \cap conv(Y) = \emptyset$. Hence, for each hyperplane H such that $H \cap C(x') \neq \emptyset$ and $H \cap conv(Y) = \emptyset$, it follows that $\delta(x(0), C(x')) = \delta(x(0), H \cap C(x'))$. □

As consequence of the preceding theorem, we conclude that to directionally round the truncated cone $C(x')$ it suffices to round the set of points defined by the intersection of the cone $C(x')$ and any hyperplane H disjoint with $conv(Y)$. In other words it is possible to reduce the dimension of the set which is directionally rounded, i.e., instead of rounding $C(x')$ it suffices to round the lower dimensional set $H \cap C(x')$. Furthermore, the set $H \cap C(x')$ may be directionally rounded very efficiently as we show in the next theorem.

Theorem 4.2. *Let H be any hyperplane such that $H \cap C(x') \neq \emptyset$ and $H \cap conv(Y) = \emptyset$. Then the set $H \cap C(x')$ can be rounded by rounding a finite number of line segments lying inside the hyperplane H.*

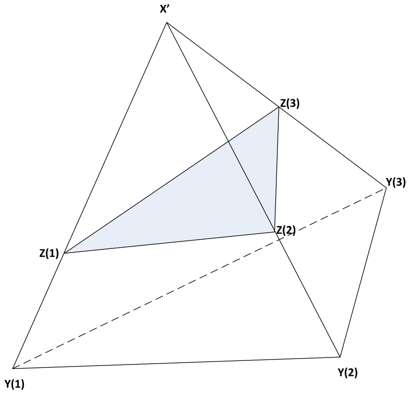

Fig. 3. Cone $C(x')$ and points z(h).

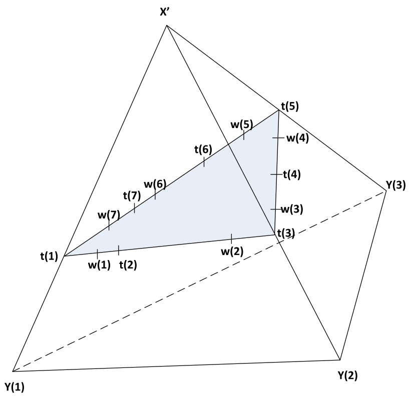

Fig. 4. Cone $C(x')$ and points from set T'.

Proof. For two points x' and x'', we define $\Delta(x', x'') = \{x(\lambda_j^*) : x(\lambda_j^*) = x' + \lambda_j^*(x'' - x'), 0 \leq \lambda_j^* = \frac{x(0)_j - x'_j}{x''_j - x'_j} \leq 1\}$. Let $z(h)$ denote the point obtained by the intersection of the hyperplane H with the edge $[x', y'(h)]$, i.e., $z(h) = [x', y'(h)] \cap H$. Further, let us denote by Z set of all $z(h)$ points, i.e., $Z = \{z(h) : h = 1, \ldots, |\overline{B}|\}$ (see Fig. 3), and define the set of points $T = Z \cup \{\Delta(z(h), z(h')) : z(h) \text{ and } z(h') \text{ are adjacent points}\}$. Index points in set T to yield $T = \{t(1), \ldots, t(p)\}$ where $p = |T|$. Then, we determine new points to be added to the set T, by randomly choosing points on each extreme edge according to the following rule. Let $[z(h), z(h')]$ be an edge and $t'(1), \ldots, t'(m)$ points on that edge determined computing $\Delta(z(h), z(h'))$ such that any open sub-edge $]t'(k), t'(k+1)[$, $k = 1, \ldots m-1$, does not contain any other $t'(j)$ point. Then the points from each edge to be added are chosen as random points $w^k \in]t'(k), t'(k+1)[$ for $k = 0, \ldots, m$ where $t'(0) = z(h)$ and $t'(m+1) = z(h')$. Adding these points to set T produces the new set denoted T' (see Fig. 4), i.e., $T' = T \cup \{w^k : k = 1, \ldots, r\}$ where $\{w^k : k = 1, \ldots, r\}$ represents set of all w^k points with respect to all edges. Therefore, we obtain $\delta(x(0), H \cap C(x')) = \delta(x(0), \{[x, y] : x, y \in T'\})$. Indeed, if $[a, b]$ is any line segment from $H \cap C(x')$ then there are points $x, y \in T'$ such that line segments $[a, b]$ and $[x, y]$ are intersected by the same hyperplane and therefore $\delta(x(0), [x, y]) = \delta(x(0), [a, b])$. More precisely, for each hyperplane $x_j = x(0)_j$, $j = 1, \ldots, n$, which intersects $H \cap C(x')$, the set $\{w^k : k = 1, \ldots, r\}$ includes points which are from different sides. For a given line segment we can easily determine another, with endpoints from $\{w^k : k = 1, \ldots, r\}$ such that both are intersected by the same hyperplane. If one or both endpoints of $[a, b]$ are in some hyperplane $x_j = x(0)_j$, then whether or not we consider these points as an intersection of $[a, b]$ and hyperplane $x_j = x(0)_j$, we will be still able to find points x, y such that line segments $[a, b]$ and $[x, y]$ are intersected by the same hyperplane. □

The previous theorems enable us to organize directional rounding of $Plane(X(\overline{B}))$ in the following way. Choose any extreme point x' of $Plane(X(\overline{B}))$ and identify plane $conv(Y)$, where $Y = \{y'(h) : h \in \overline{B}\}$ denotes the set of points $y'(h)$ such that directional rounding of all points from open sub-edge $]x', y'(h)[$ yield the same 0–1 point. Then perform directional rounding of the truncated cone $C(x')$ using Theorem 4.2 and repeat the overall procedure on $Plane(X(\overline{B})) - C(x')$. At each iteration this procedure reduces the size of a set to be examined in the next iteration, in a fashion that assures the procedure is convergent.

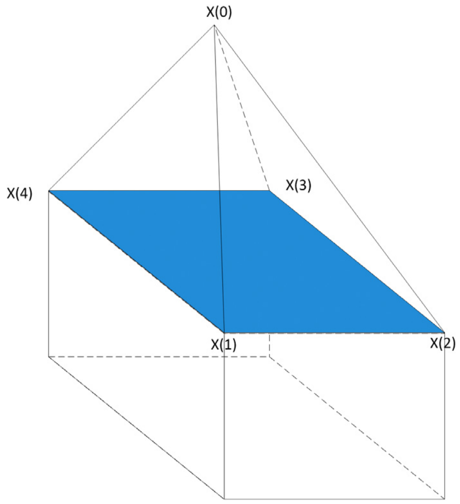

Fig. 5. Polyhedron of LP relaxation of (27).

Our convergence proof may be given an alternative interpretation. Instead of viewing it as applicable to an enhanced version of scatter search we may alternately view it as applicable to a pure IP cutting plane algorithm that uses scatter search techniques to generate intersection cuts.[1]

4.3. Illustration of convergence of scatter search

In this subsection, we give two examples to illustrate the execution of the scatter search based on directional rounding. The first example with $n = 3$ is provided in order to illustrate graphically the process of the Convergent Scatter Search. The next example with $n = 6$ is used to describe the execution of the scatter search algorithm step by step.

Example 4.3. Consider the following 0–1 mixed integer problem

$$\begin{array}{ll} \max & z = 1200x_1 + 2400x_2 + 500x_3 \\ s.t. & 2x_2 + 2x_3 \le 3 \\ & 2x_1 + 2x_2 \le 3 \\ & -2x_1 + 2x_2 \le 1 \\ & 2x_2 - 2x_3 \le 1 \\ & x_1, x_2, x_3 \in \{0, 1\}. \end{array} \tag{27}$$

The LP relaxation polyhedron of this problem is shown in Fig. 5. The optimal solution of the LP relaxation is the point $x(0) = (0.5, 1, 0.5)$ while its adjacent extreme points are $x(1) = (0, 0.5, 0)$, $x(2) = (1, 0.5, 0)$, $x(3) = (0, 0.5, 1)$ and $x(4) = (1, 0.5, 1)$. The remaining extreme points are $(0, 0, 0), (1, 0, 0), (1, 1, 0)$ and $(0, 0, 1)$. The region $Plane(X(\overline{B}))$ determined by the points $x(1), x(2), x(3)$ and $x(4)$ is colored blue in Fig. 5. Now, our procedure works as follows. First we choose any extreme point x' of $Plane(X(\overline{B}))$, say $x' = x(1)$. After that on each of edges $[x(1), x(2)]$ and $[x(1), x(4)]$ we identify points $y(1)$ and $y(2)$ closest to the $x(1)$ such that $y(1)_j = x(0)_j$ and $y(2)_k = x(0)_k$ for some j and k. In the present case the point $y(1)$ is obtained as the intersection of edge $[x(1), x(2)]$ and hyperplane $x_1 = 0.5$, while the point y_2 is the intersection point of edge $[x(1), x(3)]$ and hyperplane $x_3 = 0.5$ (see Fig. 6). The cone $C(x')$ to be directionally rounded is the triangle with vertices $x', y(1), y(2)$. Its directional rounding produces just one 0–1 point $(0, 0, 0)$. In a similar way, we determine and directionally round cones originating at $x(2), x(3), x(4)$ to obtain points $(1, 0, 0), (0, 0, 1), (1, 0, 1)$. After eliminating all cones that we directionally rounded, i.e., all parts of $Plane(X(\overline{B}))$ that we have explored, we obtain the plane presented in Fig. 7 with extreme points $x(1) = (0, 0.5, 0.5)$, $x(2) = (0.5, 0.5, 0)$, $x(3) = (1, 0.5, 0.5)$ and $x(4) = (0.5, 0.5, 1)$. Now we choose the point $x(1)$ as point x'. Its corresponding points y coincide with points $x(2)$ and $x(4)$ and therefore cone $C(x')$ is the triangle generated by the vertices $x(1)$, $x(2)$ and $x(4)$. The directional rounding of the cone $C(x')$ yields two points $(0, 0, 0)$ and $(0, 0, 1)$. Similarly, the directional rounding of the non-examined cone $C(x(3))$ produces two points $(1, 0, 0)$ and $(1, 0, 1)$. This completes our exploration. The procedure reports solution $(1, 0, 1)$ as optimal. Because of the simplicity of this problem we did not indicate the directional rounding of a cone step by step.

[1] This alternative interpretation was suggested by the observant comments of a reviewer.

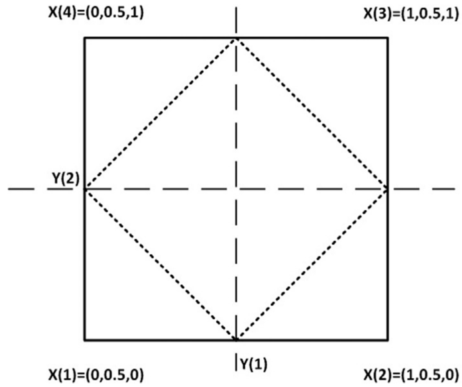

Fig. 6. Plane $X(\overline{B})$ before starting CSS — dotted line represents cones that will be cut.

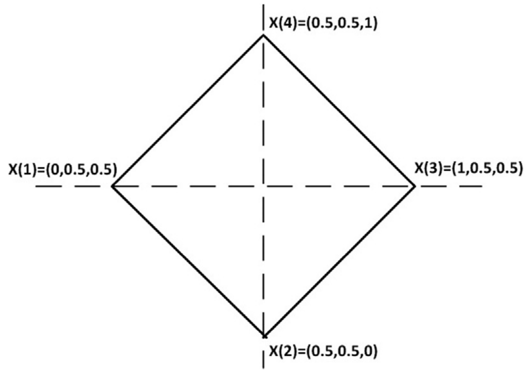

Fig. 7. Plane $X(\overline{B})$ after cutting 4 cones.

In the next example we show how the second variant of Convergent Scatter Search works.

Example 4.4.

$$\begin{array}{ll} \max & z = 100x_1 + 600x_2 + 1200x_3 + 2400x_4 + 500x_5 + 2000x_6 \\ \text{Subject To} & \\ & 8x_1 + 12x_2 + 13x_3 + 64x_4 + 22x_5 + 41x_6 \leq 80 \\ & 8x_1 + 12x_2 + 13x_3 + 75x_4 + 22x_5 + 41x_6 \leq 96 \\ & 3x_1 + 6x_2 + 4x_3 + 18x_4 + 6x_5 + 4x_6 \leq 20 \\ & 5x_1 + 10x_2 + 8x_3 + 32x_4 + 6x_5 + 12x_6 \leq 36 \\ & 5x_1 + 13x_2 + 8x_3 + 42x_4 + 6x_5 + 20x_6 \leq 44 \\ & 5x_1 + 13x_2 + 8x_3 + 48x_4 + 6x_5 + 20x_6 \leq 48 \\ & 8x_5 \leq 10 \\ & 3x_1 + 4x_3 + 8x_5 \leq 18 \\ & 3x_1 + 2x_2 + 4x_3 + 8x_5 + 4x_6 \leq 22 \\ & 3x_1 + 2x_2 + 4x_3 + 8x_4 + 8x_5 + 4x_6 \leq 24 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \in \{0, 1\} \end{array} \tag{28}$$

The optimal solution of the LP relaxation is the point $x(0) = (0, 0, 1, 0.36, 0.12, 1)$ with an objective function value 4134.74. The extreme points adjacent to $x(0)$ are the following 6 points:

$x(1) = (1, 0, 1, 0.25, 0.10, 1)$
$x(2) = (0, 0.92, 1, 0, 0.68, 1)$
$x(3) = (0, 0, 0, 0.54, 0.19, 1)$
$x(4) = (0, 0, 1, 0.59, 0.65, 0.36)$
$x(5) = (0, 0, 1, 0.38, 0, 1)$
$x(6) = (0, 0, 1, 0.06, 1, 1).$

These points are contained in the hyperplane P given by the equation $0.1x_1 + 0.18x_2 + 0.22x_3 + x_4 + 0.32x_5 + 0.65x_6 = 1.26$. The optimal vertex in P w.r.t. to the objective function is the point $x(5)$ while remaining extreme points are adjacent to it. The $y(h)$ points on each edge $[x(5), x(h)]$ for $h \in \{1, 2, 3, 4, 6\}$ are the following 5 points:

$$y(1) = (0.13, 0, 1, 0.36, 0.01, 1)$$
$$y(2) = (0, 0.04, 1, 0.36, 0.03, 1)$$
$$y(3) = (0, 0, 0.33, 0.49, 0.12, 1)$$
$$y(4) = (0, 0, 1, 0.42, 0.12, 0.88)$$
$$y(5) = (0, 0, 1, 0.36, 0.06, 1)$$

while the best solution obtained by directionally rounding the cone originating at $x(5)$ and bounded by the face which contains points $y(h)$ is the point $(0, 0, 0, 1, 0, 0)$ with an objective function value of 2400. The hyperplane H containing points $y(h)$ and $x(0)$ is defined as $0.19x_3 + x_4 + 0.480x_6 = 1.03$.

In the second iteration we repeat the overall procedure on the new problem derived from the starting problem by imposing two cuts, one defined by hyperplane P which excludes the point $x(0)$ and another defined by hyperplane H which excludes the point $x(5)$. The optimal solution of the new problem is the point $x(0) = (0, 0.04, 1, 0.36, 0.03, 1)$ with an objective function value 4113.17 and its adjacent extreme points are:

$$x(1) = (0.14, 0, 1, 0.36, 0.01, 1)$$
$$x(2) = (0, 0, 0.33, 0.49, 0.13, 1)$$
$$x(3) = (0, 0, 1, 0.42, 0.13, 0.88)$$
$$x(4) = (0, 0, 1, 0.36, 0.06, 1)$$
$$x(5) = (0, 0.06, 1, 0.36, 0, 1)$$
$$x(6) = (0, 0.91, 1, 0, 0.68, 1)$$

while the hyperplane P that contains them is defined as $0.077x_1 + 0.23x_2 + 0.21x_3 + x_4 + 0.24x_5 + 0.61x_6 = 1.20$. The optimal vertex in P w.r.t. to the objective function is the point $x(5)$ while the remaining extreme points are adjacent to it. The $y(h)$ points on each edge $[x(5), x(h)]$ for $h \in \{1, 2, 3, 4, 6\}$ are:

$$y(1) = (0, 0.04, 0.83, 0.39, 0.032, 1)$$
$$y(2) = (0, 0.04, 1, 0.38, 0.03, 0.97)$$
$$y(3) = (0, 0.04, 1, 0.36, 0.01, 1)$$
$$y(4) = (0, 0.10, 1, 0.35, 0.03, 1)$$
$$y(5) = (0.03, 0.04, 1, 0.36, 0.00, 1).$$

The best solution obtained by directionally rounding the cone $C(x(5))$ is the point $(0, 1, 1, 0, 0, 1)$ with an objective function value of 3800 which corresponds to the optimal solution of the starting problem. The hyperplane H which contains the $y(h)$ points is given as $0.31x_2 + 0.19x_3 + x_4 + 0.48x_6 = 1.04$. The overall procedure is then repeated on the new problem adding two cuts as in the previous iteration. It should be emphasized that this small example discloses our convergence method can consume a long time to establish optimality in spite of finding an optimal solution in the second iteration. Therefore, we do not illustrate all iterations required to prove optimality.

5. Conclusions

In this paper, the Convergent Scatter Search with directional rounding for solving 0–1 MIP problems is introduced for the first time. The idea for using Scatter search as a method for exploring points on the imposed cutting plane is justified by proving theorems which show that the cutting plane contains a polyhedral region which produces all feasible 0–1 solutions by directional rounding. We also provide an efficient method to carry out directional rounding relative to a line segment. To illustrate our main ideas, we demonstrate the operation of Convergent Scatter Search in two small examples. Additionally, we identify two variants for implementing the method.

Future work will include embedding these variants in a branch and bound framework, taking advantage of the fact that the proposed methods can be easily parallelized to execute several tasks simultaneously. Another avenue for future work is to develop heuristic approaches based on the results developed to support the proposed algorithms.

Acknowledgments

We thank the annonymous referees for valuable comments that helped us to improve the paper.

References

[1] R. Aboudi, A. Hallefjord, R. Helming, K. Jornsten, A note on the pivot and complement heuristic for 0–1 programming problems, Oper. Res. Lett. 8 (1989) 21–23.
[2] T. Achterberg, T. Berthold, Improving the feasibility pump, Discrete Optim. 4 (1) (2007) 77–86.
[3] E. Balas, The intersection cut–A new cutting plane for integer programming, Oper. Res. 19 (1971) 19–39.
[4] E. Balas, S. Ceria, G. Cornuéjols, A lift-and-project cutting plane algorithm for mixed 0–1 programs, Math. Program. 58 (1993) 295–324.
[5] E. Balas, S. Ceria, M. Dawande, F. Margot, G. Pataki, OCTANE: A new heuristic for pure 0-1 programs, Oper. Res. 49 (2001) 207–225.
[6] E. Balas, F. Margot, Generalized intersection cuts and a new cut generating paradigm, Math. Program. 137 (2013) 19–35.
[7] E. Balas, C. Martin, Pivot and complement-a heuristic for 0–1 programming, Manag. Sci. 26 (1980) 86–96.
[8] E. Balas, S. Schmieta, C. Wallace, Pivot and shift-a mixed integer programming heuristic, Discrete Optim. 1 (2004) 3–12.
[9] L. Bertacco, M. Fischetti, A. Lodi, A feasibility pump heuristic for general mixed-integer problems, Discrete Optim. 4 (1) (2007) 63–76.
[10] J. Brimberg, P. Hansen, N. Mladenović, Attraction probabilities in variable neighborhood search, 4OR 8 (2010) 181–194.
[11] E. Danna, E. Rothberg, C. Le Pape, Exploring relaxation induced neighborhoods to improve MIP solutions, Math. Program. 102 (1) (2005) 71–90.
[12] G. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, 1963.
[13] J. Eckstein, M. Nediak, Pivot, cut and dive: a heuristic for 0-1 mixed integer programming, J. Heuristics 13 (2007) 471–503.
[14] M. Fischetti, F. Glover, A. Lodi, The feasibility pump, Math. Program. 104 (1) (2005) 91–104.
[15] M. Fischetti, A. Lodi, Local branching, Math. Program. 98 (2003) 23–47.
[16] F. Glover, A multiphase dual algorithm for the zero-one integer programming problem, Oper. Res. 13 (1965) 879–919.
[17] F. Glover, Cut search methods in integer programming, Math. Program. 3 (1972) 86–100.
[18] F. Glover, Heuristics for integer programming using surrogate constraints, Decis. Sci. 8 (1977) 156–166.
[19] F. Glover, Scatter search and star-paths: beyond the genetic metaphor, OR Spektrum 17 (1995) 125–137.
[20] F. Glover, A template for scatter search and path relinking, in: J.K. Hao, E. Lutton, E. Ronald, M. Schoenauer, D. Snyers (Eds.), Artificial Evolution, in: Lecture Notes in Computer Science, Springer-Verlag, New York, 1998, pp. 1–51.
[21] F. Glover, Scatter search and path relinking, in: D. Corne, M. Dorigo, F. Glover (Eds.), New Ideas in Optimisation, McGraw-Hill, Boston, 1999.
[22] F. Glover, S. Hanafi, Tabu search and finite convergence, Discrete Appl. Math. 119 (2002) 3–36.
[23] F. Glover, M. Laguna, General purpose heuristics for integer programming-Part II, J. Heuristics 3 (1997) 161–179.
[24] F. Glover, M. Laguna, Fundamentals of scatter search and path relinking, Control Cybernet. 29 (2000) 653–684.
[25] F. Glover, M. Laguna, R. Marti, Scatter search, in: A. Ghosh, S. Tsutsui (Eds.), Theory and Applications of Evolutionary Computation: Recent Trends, Springer-Verlag, New York, 2000.
[26] M.A. González, C.R. Vela, R. Varela, Scatter search with path relinking for the flexible job shop scheduling problem, European J. Oper. Res. 245 (1) (2015) 35–45.
[27] S. Hanafi, On the convergence of tabu search, J. Heuristics 7 (2001) 47–58.
[28] S. Hanafi, J. Lazić, N. Mladenović, Variable neighbourhood pump heuristic for 0–1 mixed integer programming feasibility, Electron. Notes Discrete Math. 36 (2010) 759–766.
[29] S. Hanafi, J. Lazić, N. Mladenović, C. Wilbaut, I. Crévits, Hybrid variable neighbourhood decomposition search for 0–1 mixed integer programming problem, Electron. Notes Discrete Math. 36 (2010) 883–890.
[30] S. Hanafi, C. Wilbaut, Improved convergent heuristics for the 0-1 multidimensional knapsack problem, Ann. OR 183 (2011) 125–142.
[31] P. Hansen, N. Mladenović, D. Urosević, Variable neighborhood search and local branching, Comput. Oper. Res. 33 (10) (2006) 3034–3045.
[32] Z. Khooban, R.Z. Farahani, E. Miandoabchi, W. Szeto, Mixed network design using hybrid scatter search, European J. Oper. Res. 247 (3) (2015) 699–710.
[33] M. Laguna, R. Marti, Scatter Search–Methodology and Implementations in C, Kluwer Academic Publishers, Norwell, MA, 2003.
[34] M. Laguna, R. Martí, Experimental testing of advanced scatter search designs for global optimization of multimodal functions, J. Global Optim. 33 (2) (2005) 235–255.
[35] J.B. Lasserre, An analytical expression and an algorithm for the volume of a convex polyhedron in R^n, Journal of optimization theory and applications 39 (3) (1983) 363–377.
[36] J. Lazić, S. Hanafi, N. Mladenović, D. Urosević, Variable neighbourhood decomposition search for 0–1 mixed integer programs, Comput. Oper. Res. 37 (2010) 1055–1067.
[37] J. Lazić, R. Todosijević, S. Hanafi, N. Mladenović, Variable and Single Neighbourhood Diving for MIP Feasibility, Yugoslav Journal of Operations Research, 2014, http://dx.doi.org/10.2298/YJOR140417027L.
[38] A. Lokketangen, F. Glover, Solving zero-one mixed integer programming problems using tabu search, European J. Oper. Res. 106 (1998) 624–658.
[39] R. Martí, A. Corberán, J. Peiró, Scatter search for an uncapacitated p-hub median problem, Comput. Oper. Res. 58 (2015) 53–66.
[40] R. Marti, F. Glover, M. Laguna, Principles of scatter search, European J. Oper. Res. 169 (2006) 359–372.
[41] B. Naderi, R. Ruiz, A scatter search algorithm for the distributed permutation flowshop scheduling problem, European J. Oper. Res. 239 (2) (2014) 323–334.
[42] M.G.C. Resende, C.C. Ribeiro, F. Glover, R. Marti, Scatter search and path-relinking: Fundamentals, advances and applications, in: Handbook of Metaheuristics, in: International Series in Operations Research & Management Science, vol. 146, 2010, pp. 87–107.
[43] R. Todosijević, S. Hanafi, F. Glover, Scatter search and star paths with directional rounding for 0–1 mixed integer programs: Heuristic algorithms. working paper.
[44] H. Tuy, Concave programming under linear constraints, Sov. Math. (1964) 1437–1440.
[45] C. Wilbaut, S. Hanafi, New convergent heuristics for 0–1 mixed integer programming, European J. Oper. Res. 195 (2009) 62–74.
[46] L.A. Wolsey, G.L. Nemhauser, Integer and combinatorial optimization, 55, John Wiley & Sons, 1999.
[47] R.D. Young, Hypercylindrically deduced cuts in zero-one integer programs, Oper. Res. 19 (6) (1971) 1393–1405.