

VARIABLE NEIGHBORHOOD SEARCH

Prepared by :

BENZAKRY Aimad
KHARBAB Achraf
MACHI Imad

Supervised by :

Pr. Ouaarab Aziz

CONTENT

- 01** INTRODUCTION
- 02** VNS DEFINITION
- 03** APPROACH FAMILY
- 04** VNS COMPONENTS
- 05** VNS ALGORITHM
- 06** INTENSIFICATION & DIVERSIFICATION
- 07** VNS VARIANTS
- 08** APPLICATIONS
- 09** ADVANTAGES & LIMITS
- 10** LAB
- 11** CONCLUSION

INTRODUCTION

PROBLÈMES D'OPTIMIZATIONS:

Considérons un problème d'optimisation combinatoire ou globale:

$$\min\{ f(x) | x \in X \subseteq S \},$$

où $f(x)$ est la fonction objectif à minimiser, S est l'espace de solutions et X est l'ensemble des solutions faisables.

Une solution $x^* \in X$ est optimale si :

$$f(x^*) < f(x), \forall x \in X$$

PROBLÉMATIQUE:

Le cas du **problème du voyageur de commerce**, par exemple, l'espace de recherche croit en $(n-1)!$, où n est le nombre de villes à visiter, ce qui dépasse rapidement les capacités de calcul de n'importe quel ordinateur.

Avec seulement 50 villes, il faudra évaluer $49!$ trajets, soit $6,08 \cdot 10^{62}$
C'est l'explosion combinatoire.

LES HEURISTIQUES:

- **NP-difficiles:** aucun algorithme avec un nombre d'étapes polynomial par rapport à la taille des instances n'est connu, et que trouver un tel algorithme pour l'un de ces problèmes équivaudrait à en obtenir un pour tous.
- Les **heuristiques** fournissent rapidement une solution approximative, et parfois même une solution optimale, Mais sans preuve de son optimalité.

LES HEURISTIQUES:

- **Local optima:**

Un optimum local x_l est tel que:

$$f(x_l) < f(x), \forall x \in N(x_l) \cap X$$

où $N(x_l)$: le voisinage de x_l

- Les **heuristiques** servent deux objectifs principaux:

- éviter un temps de calcul excessif,
- aborder le problème : les optimas locaux.

POURQUOI LES METAHEURISTIQUES ?

- Problèmes:
 - Risque de convergence vers un optimum local,
 - Performance sensible à la configuration,
 - Peu d'adaptabilité à différents problèmes
- la valeur optimale globale $f(x^*)$ peut différer considérablement de la valeur moyenne du minimum local, voire même de la meilleure valeur parmi plusieurs obtenue par une heuristique simple (phénomène appelé la **catastrophe de Tchebycheff**).

METAHEURISTIQUE - VNS:

- **Les métahéuristiques** sont un framework pour construire des heuristiques destinées à résoudre des problèmes d'optimisation combinatoire et globale.
- Exploration de l'espace des solutions
- Évitement des optima locaux
- Adaptabilité

RECHERCHE A VOISINAGE VARIABLE (VNS) :

- La **Recherche à Voisinages Variables** (RVV) a été proposé par Mladenovic et Hansen en 1997.
- exploiter le changement systématique des voisinages
- Soit $L = (N(1), \dots, N(T))$ une liste finie de voisinages, où $N(t)(s)$ est l'ensemble des solutions dans le voisinage de s . Dans la plupart des méthodes de recherche locale, on a $T=1$.
- En général, on change de voisinage à chaque fois qu'un voisinage ne permet plus d'améliorer la solution initiale après avoir fait une recherche locale dans ce voisinage.

DOCUMENTATION DU PROBLEME A RESOUDRE:

(i) Évaluation de la difficulté :

- Est-ce un problème NP-difficile ?
- S'il est dans P, quelle est la complexité de l'algorithme exact le mieux connu, et est-elle suffisamment basse pour que des instances réalistes puissent être résolues en un temps raisonnable ?

(ii) Évaluation des performances des algorithmes précédents :

- Existe-t-il des instances de données disponibles pour le problème?
- Quelles sont les plus grandes instances qui ont été résolues exactement ?

DOCUMENTATION DU PROBLEME A RESOUDRE:

iii) Évaluation des performances des heuristiques précédentes :

- Quelles métaheuristiques ont été appliquées à ce problème ?
- Quelles sont les performances des heuristiques résultantes en termes de taille des problèmes résolus, d'erreur et de temps de calcul ?

(iv) Quelles étapes sont utilisées dans les heuristiques proposées ?

- Quels sont les voisinages correspondants de la solution actuelle ?
- Des codes pour ces heuristiques sont-ils disponibles ?
- Des codes pour des méthodes de descente simples sont-ils disponibles ?

VNS DEFINITION

VNS DEFINITION

VNS, introduced by Mladenovic and Hansen in 1997, is a powerful metaheuristic designed to address optimization problems by utilizing the concept of neighborhood change. It efficiently navigates the optimization landscape using several key observations

VNS DEFINITION:

In the process of searching for a global min, given an initial solution s , a neighboring solution s' is generated from $N_1(s)$, and a local search procedure is applied to obtain a solution s'' . If $f(s'') < f(s)$, then $s = s''$ and a new neighboring solution is generated in $N_1(s)$. Otherwise, the current solution remains s and a new solution s' is generated in $N_2(s)$.

More generally, a change of neighborhood occurs whenever one of them, after the application of the local search procedure, fails to improve the current solution s . However, as soon as a neighborhood leads to an improvement of s , the process restarts with the first neighborhood.

NEIGHBORHOOD CHANGE FACTS:

- **Fact 1:** A local minimum in one neighborhood is not necessarily a minimum in another.
- **Fact 2:** A global minimum is a local minimum in all possible neighborhoods.
- **Fact 3:** Local minima within different neighborhoods are often relatively close to each other.

V N D

VARIABLE NEIGHBORHOOD DESCENT (VND)

Variable Neighborhood Descent (VND) is a steepest descent heuristic, also known as the best improvement local search. It involves selecting an initial solution x , identifying the steepest descent direction within a neighborhood $N(x)$, and moving towards the minimum of the objective function $f(x)$ within that neighborhood. If no descent direction is found, the heuristic stops; otherwise, it is iterated.

The background features two sets of thin, black, wavy lines. One set originates from the bottom left corner and curves upwards towards the top right. The other set originates from the top right corner and curves downwards towards the bottom left.

RVNS

REDUCED VARIABLE NEIGHBORHOOD SEARCH (RVNS):

RVNS is an extension of Variable Neighborhood Search that systematically explores nested neighborhoods around a current point. It intelligently chooses directions and depths to uncover better solutions, leveraging the implicit information present in the vicinity of local optima



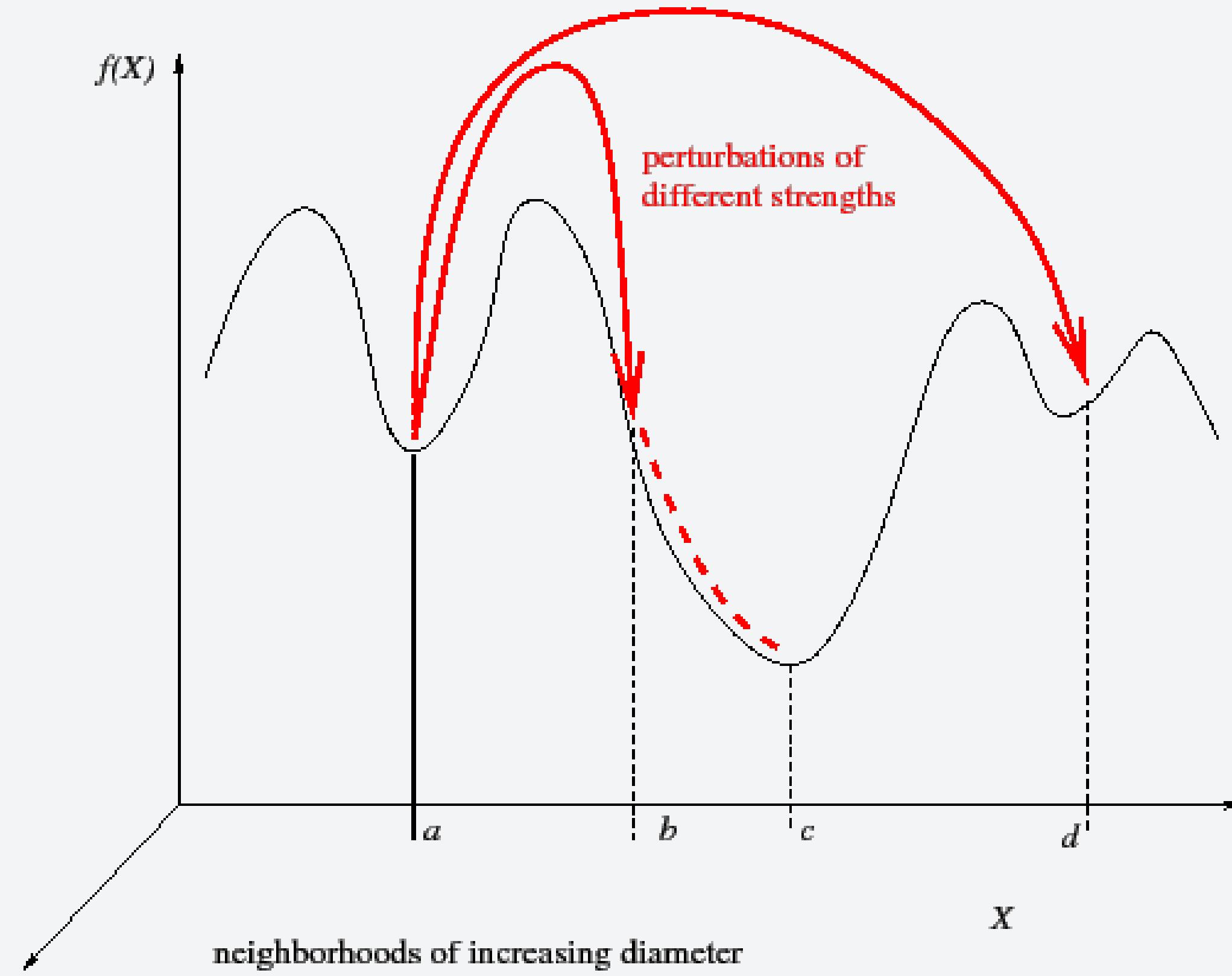
A rectangular frame with a black border is centered on the page. Inside the frame, the words "APPROACH FAMILY" are written in a bold, black, sans-serif font, with a bounding box of approximately [450, 230, 650, 750].

APPROACH FAMILY

LOCAL SEARCH

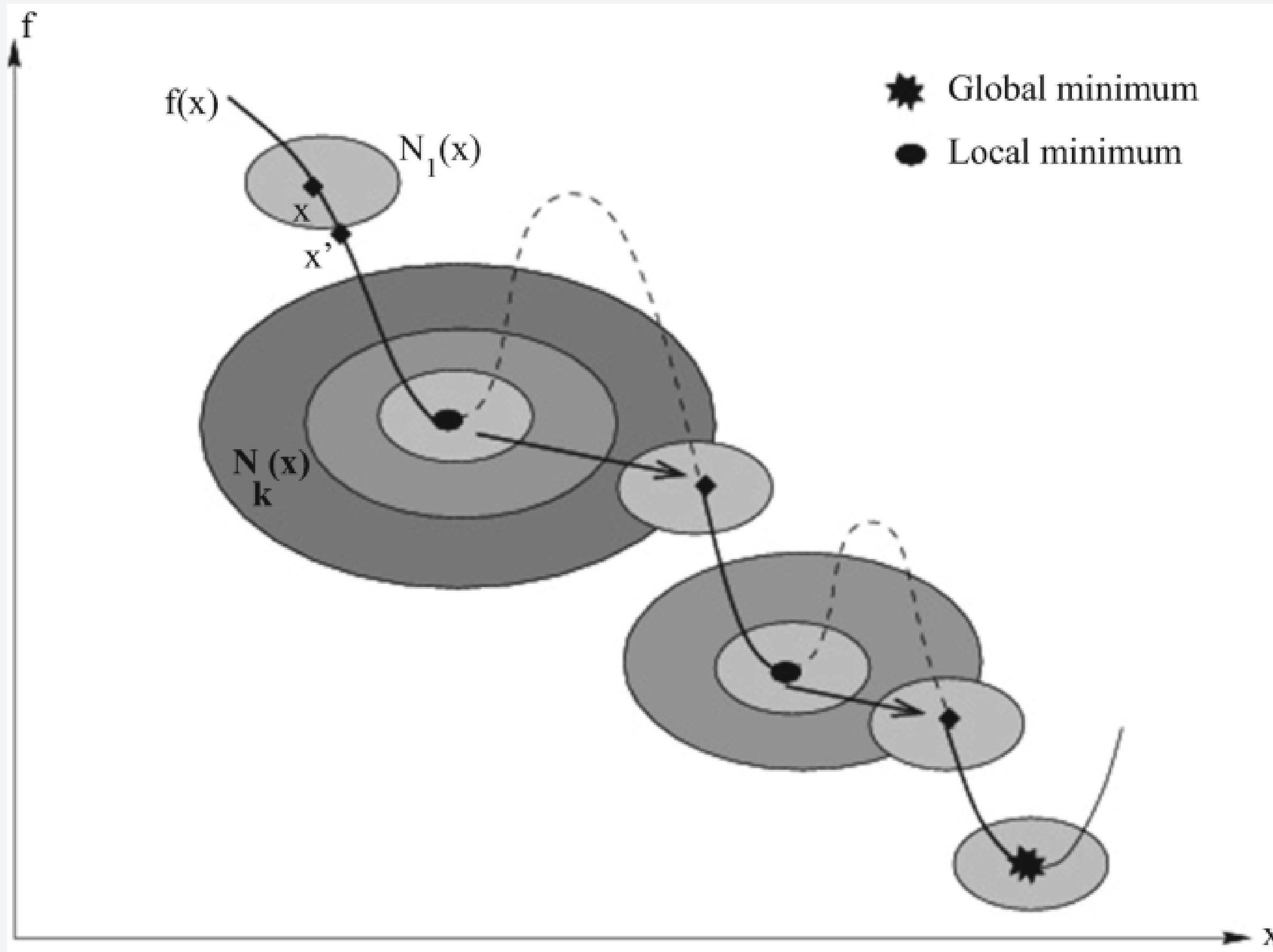
Local search is a smart algorithmic strategy crafted for optimization challenges, aiming to pinpoint the best solution among a set of options. It operates by smoothly moving from one solution to a nearby one within the candidate solution space. This exploration persists until an optimal solution emerges or a predetermined stopping point is reached. Notably, the decision-making process hinges solely on information from neighboring solutions, earning it the label 'local search.' This method offers an effective means of navigating solution spaces to uncover optimal outcomes.

LOCAL SEARCH



WHY VNS IS LOCAL SEARCH ALGORITHM

- 1. Neighborhood Exploration:** VNS actively explores and exploits the information embedded in different neighborhood structures. It shifts from one neighborhood to another, seeking improved solutions within each.
- 2. Incumbent Solution Update:** Similar to local search heuristics, VNS updates its incumbent solution based on the selected, better solution found within the current neighborhood. This process continues until the algorithm converges to a solution that serves as a local optimum within its specific neighborhood structure.
- 3. Flexibility in Search Strategies:** VNS accommodates various local search strategies, such as the first improvement and best improvement, allowing it to adapt to the specific characteristics of the optimization problem at hand.



VNS COMPONENTS

COMPOSANTS DE VNS:

- (1) SOLUTION INITIALE
- (2) STRUCTURES DES VOISINAGES
- (3) MÉTHODE DE RECHERCHE LOCALE
- (4) DÉPLACEMENT
- (5) CRITÈRE D'ARRÊT

SOLUTION INITIALE:

- La sélection de la solution de départ influence la performance globale.
- Impact sur la convergence et la qualité de la solution finale.
- Générée aléatoirement
- Obtenue à l'aide d'un algorithme glouton

STRUCTURES DES VOISINS:

- Soit $N(x)$ désignant une structure de voisinage d'une solution donnée x .
En général, un voisinage $N(x)$ est défini par rapport à une fonction métrique δ introduite dans l'espace des solutions S comme suit :
$$N(x) = \{y \in X | \delta(x, y) \leq a\},$$
 où a est un nombre positif donné.

MÉTHODE DE RECHERCHE LOCALE:

- Exploration du voisinage : On cherche le meilleur voisin x'' de x (On applique une méthode de recherche locale avec x la solution initiale)
- First improvement Local search
- Best improvement Local search

DÉPLACEMENT:

- Secousse : on génère un point x' au hasard dans le i -ième voisinage de x ($x' \in N_i(x)$)
- Si l'optimum local x'' est meilleur que le point d'origine x , alors on s'y déplace , et on reprend la recherche dans $N_1(x)$, sinon on se déplace vers le voisinage suivant.

CRITÈRE D'ARRÊT:

- Le critère d'arrêt de la recherche locale peut être une limite en durée. Une autre possibilité est de s'arrêter quand la meilleure solution trouvée par l'algorithme n'a pas été améliorée depuis un nombre donné d'itérations.

VNS ALGORITHM

VNS INGREDIENTS

- (1) SHAKING PROCEDURE.
- (2) IMPROVEMENT PROCEDURE.
- (3) NEIGHBORHOOD CHANGE STEP.

1) SHAKING PROCEDURE

- Predefined Set of Neighborhood Operators (N)
- Objective: Escape Local Minima traps
- Solution Selection: Random
- Order of Neighborhood Structures

2) IMPROVEMENT PROCEDURES USING LOCAL SEARCH

- First Improvement Strategy
- Best Improvement Strategy
- Variable Neighborhood Descent (VND)

FIRST IMPROVEMENT STRATEGY

Function LS_FI(x, N)

repeat

 Let $N(x) = \{x^1, \dots, x^p\}$;

$i \leftarrow 0$;

$x' \leftarrow x$;

repeat

$i \leftarrow i + 1$;

if $f(x^i) < f(x)$ **then**

$x \leftarrow x^i$;

break;

end

until $i = p$;

until $f(x') \leq f(x)$;

return x' ;

BEST IMPROVEMENT STRATEGY

```
Function LS_BI( $x$ ,  $N$ )
repeat
     $x' \leftarrow x$ ;
     $x \leftarrow \operatorname{argmin}_{y \in N(x')} f(y)$ ;
until  $f(x') \leq f(x)$ ;
return  $x'$ ;
```

VARIABLE NEIGHBORHOOD DESCENT (VND)

Function B-VND(x, ℓ_{max}, N)

repeat

$stop \leftarrow false;$

$\ell \leftarrow 1;$

$x' \leftarrow x;$

repeat

$x'' \leftarrow \operatorname{argmin}_{y \in N_\ell(x)} f(y);$

 Neighborhood_change_sequential(x, x'', ℓ);

until $\ell = \ell_{max}$;

if $f(x') \leq f(x)$ **then**

$stop \leftarrow true;$

end

until $stop = true$;

return x' ;

3) NEIGHBORHOOD CHANGE STEP

- Sequential Neighborhood Change Step
- Cyclic Neighborhood Change Step
- Skewed Neighborhood Change Step

SEQUENTIAL NEIGHBORHOOD CHANGE STEP

```
if  $f(x') < f(x)$  then  
    |  $x \leftarrow x'$ ;  
    |  $k \leftarrow 1$ ;  
else  
    |  $k \leftarrow k + 1$ ;  
end
```

CYCLIC NEIGHBORHOOD CHANGE STEP

```
k ← k + 1;  
if f(x') < f(x) then  
    | x ← x';  
end
```

SKEWED NEIGHBORHOOD CHANGE STEP

```
if  $f(x') - f(x) < \alpha d(x', x)$  then
|    $x \leftarrow x'$ ;
|    $k \leftarrow 1$ ;
else
|    $k \leftarrow k + 1$ ;
end
```



INTENSIFICATION & DIVERSIFICATION

DEFINITION

Intensification and **diversification** are two important concepts in optimization algorithms, including Variable Neighborhood Search.

- **Intensification:** refers to the process of focusing the search on promising areas of the solution space, typically by exploiting information gained from previous iterations or solutions.
- **Diversification:** on the other hand, refers to the process of exploring new areas of the solution space, typically by introducing randomness or perturbations to the current solution.

RM: Balancing intensification and diversification is important for achieving good performance in optimization algorithms

HOW INTENSIFICATION AND DIVERSIFICATION IMPLEMENTED IN VNS ?

In Variable Neighborhood Search (VNS), the trade-off between intensification and diversification is balanced in the **shaking procedure**.

- The **shaking procedure** involves checking how sensitive the objective function is to small changes (shakes) in the solution, and adjusting the neighborhood structure accordingly. For some problem instances, a completely random jump in the neighborhood is too diversified, so an intensified shaking procedure can be used to increase the intensification of the search. For example, a k-interchange neighborhood may be reduced by repeating k times random add followed by best drop moves.

The basic and general schemes of VNS combine Variable Neighborhood Descent (VND), which is mostly or entirely deterministic and focuses on intensification, with Reduced Variable Neighborhood Search (RVNS), which is stochastic and focuses on diversification.

By choosing different sets of neighborhood structures, one can adjust the balance between intensification and diversification. Additionally, **the stopping condition** can also be used to control the balance between intensification and diversification.

For example, stopping the search after a fixed number of iterations will tend to emphasize diversification, while stopping the search after a certain level of improvement is achieved will tend to emphasize intensification.



VNS VARIANTS

1) BASIC VNS

```
Function Basic_VNS( $x, k_{max}, \mathcal{N}, N$  )  
repeat  
    |  $k \leftarrow 1;$   
    | while  $k \leq k_{max}$  do  
    |     |  $x' \leftarrow \text{Shake}(x, k, \mathcal{N});$   
    |     |  $x'' \leftarrow \text{Local\_search}(x', N);$   
    |     |  $\text{Neighborhood\_change\_sequential}(x, x'', k);$   
    | end  
until stopping condition is fulfilled;  
return  $x;$ 
```

2) REDUCED VNS

```
Function Reduced_VNS( $x, k_{max}, \mathcal{N}$ )
repeat
     $k \leftarrow 1;$ 
    while  $k \leq k_{max}$  do
         $x' \leftarrow \text{Shake}(x, k, \mathcal{N});$ 
        Neighborhood_change_sequential( $x, x', k$ );
    end
until stopping condition is fulfilled;
return  $x$ ;
```

3) GENERAL VNS

```
Function General_VNS( $x, k_{max}, \ell_{max}, \mathcal{N}, N$ )
repeat
     $k \leftarrow 1;$ 
    while  $k \leq k_{max}$  do
         $x' \leftarrow \text{Shake}(x, k, \mathcal{N});$ 
         $x'' \leftarrow \text{VND}(x', \ell_{max}, N);$ 
        Neighborhood_change_sequential( $x, x'', k$ );
    end
until stopping condition is fulfilled;
return  $x$ ;
```

4) NESTED VNS

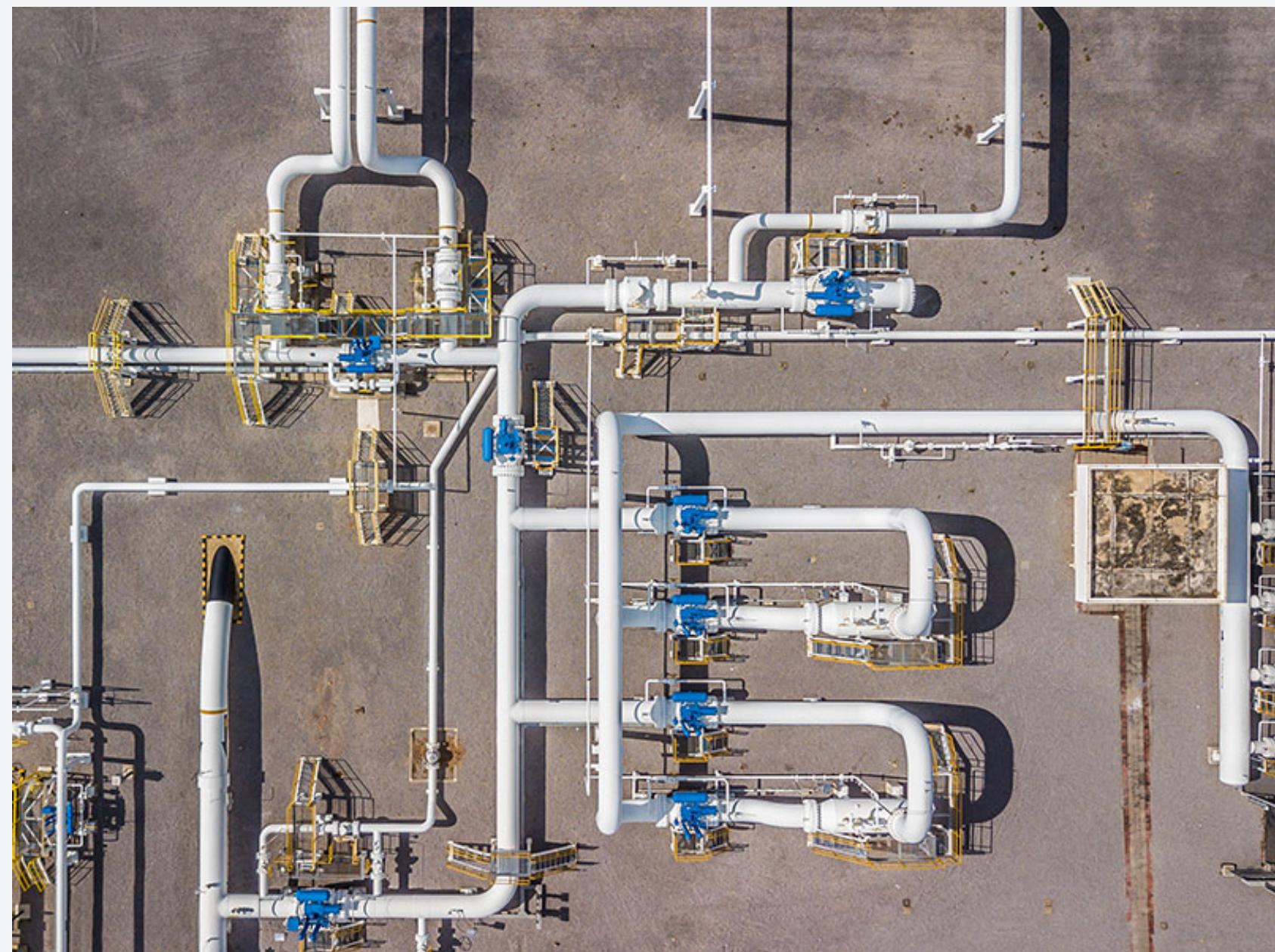
```
Function Nested_VNS( $x, b, k_{max}, \ell_{max}, \mathcal{N}, \mathcal{N}', \mathcal{N}''$ )
 $N^* = N'_1 \circ N'_2 \circ \dots \circ N'_b;$ 
 $x' \leftarrow x;$ 
repeat
     $stop = true;$ 
     $x \leftarrow x';$ 
    for each  $y \in N^*(x)$  do
         $x'' \leftarrow \text{VNS}(y, k_{max}, \ell_{max}, \mathcal{N}, \mathcal{N}'')$ ;
        if  $f(x'') < f(x')$  then
             $stop = false;$ 
             $x' \leftarrow x'';$ 
        end
    end
    until  $stop = true$ ;
return  $x'$ ;
```



APPLICATIONS

PIPELINE DESIGN PROBLEM:

In Gabon, a VNS (Variable Neighborhood Search) approach was employed to optimize the design of an oil pipeline. The goal was to discover the most cost-effective route that could efficiently transport the necessary volume of oil. VNS systematically explored various solution spaces, yielding superior results compared to other optimization methods.



OPTIMIZING TRANSPORTATION AND ROUTING OF MEDICAL RESOURCES

VNS dynamically adapts to evolving constraints, iteratively adjusting routes and resource allocations. By intelligently navigating the solution space, it swiftly identifies optimal paths, minimizing travel time and ensuring efficient resource utilization in healthcare logistics.





ADVANTAGES & LIMITS

AVANTAGES:

- Elle est simple à utilisée puisqu'elle est basée sur un principe simple qui nous permet de changer systématiquement de voisinage lorsqu'on se retrouve bloqué dans un minimum local.
- Etant très généraliste, elle s'applique à un grand nombre de problèmes d'optimisation combinatoire.
- Elle est facile à utiliser : les différentes étapes de l'algorithme sont faciles à comprendre et à mettre en œuvre.

AVANTAGES:

- Elle est efficace : les meilleures solutions sont obtenues en un temps de calcul modéré.
- Le fait d'utiliser plusieurs voisinages permet de diversifier l'exploration de l'espace de solution afin d'accéder à un plus grand nombre de régions intéressantes, ce qui conduit à une méthode plus robuste que le recuit simulé ou la recherche tabou.

LIMITES:

- Elle est souvent moins puissante que des méthodes exactes sur certains types de problèmes.
- Elle ne garantie pas non plus la découverte d'un optimum global en un temps fini.
- un mauvais choix de la solution initiale, les structures des voisinage, le critere d'arret peut amener a des solutions moins proches du min global ou un temps d execution plus long.



LAB



CONCLUSION

- Les métaheuristiques étant très généralistes, elles peuvent être adaptées à tout type de problème d'optimisation pouvant se réduire à une « boîte noire ». Elles sont souvent moins puissantes que des méthodes exactes sur certains types de problèmes. Elles ne garantissent pas non plus la découverte de l'optimum global en un temps fini. Cependant, un grand nombre de problèmes réels n'est pas optimisable efficacement par des approches purement mathématiques, les métaheuristiques peuvent alors être utilisées avec profit.

References:

- Hansen, P., Mladenović, N., Todosijević, R., & Hanafi, S. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, 5(3), 423-454.
- Mladenović, N., & Hansen, P. (1997). Variable neighborhood search. *Computers & Operations Research*, 24(11), 1097-1100.