

Carthagène

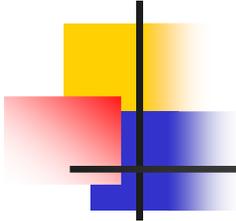


Voyager à l'aide de l'optimisation
combinatoire

Simon de Givry

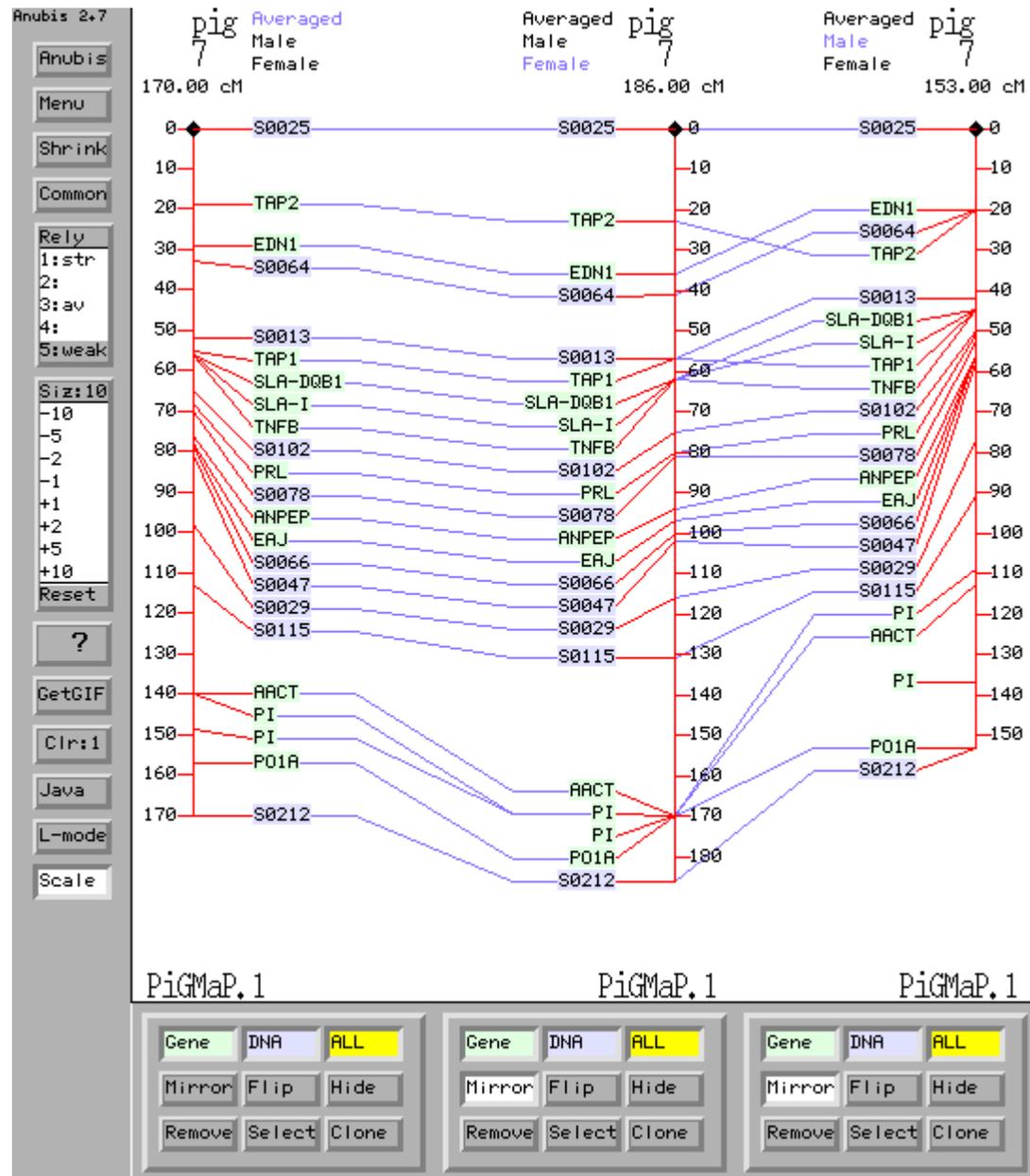


Trouver un tour passant par toutes les villes une seule fois et ayant une distance totale minimale



Matrice des distances (km)

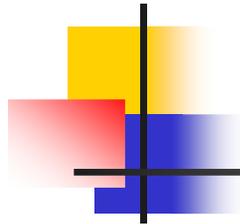
Distances	Camara	Canico	Funchal	...
Camara	0	15	7	...
Canico	15	0	8	...
Funchal	7	8	0	...
...



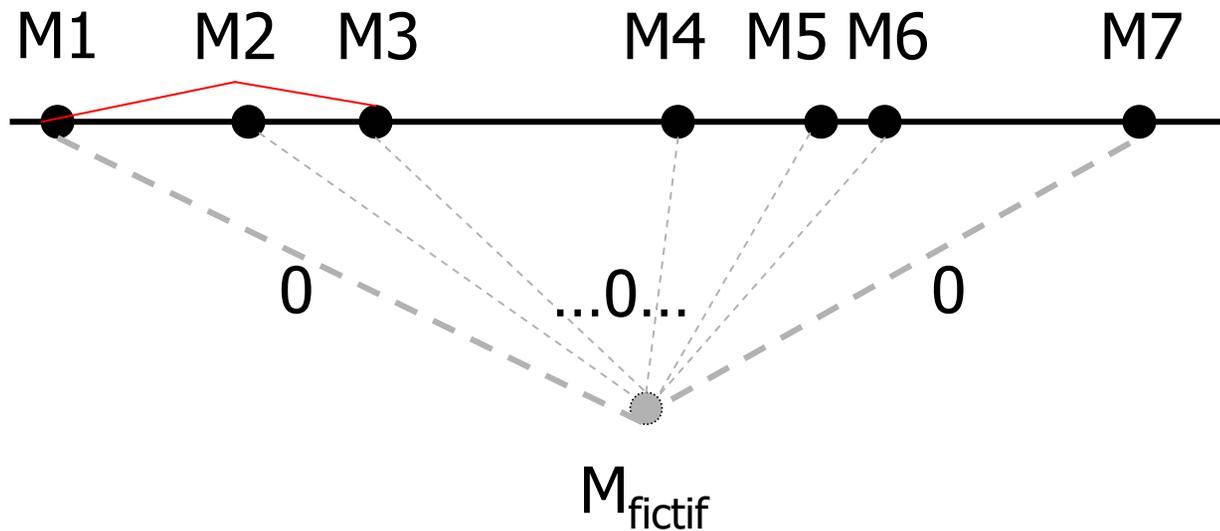
Trouver un ordre des marqueurs qui maximise la vraisemblance

Matrice des distances 2-points (Haldane)

Distances	M1	M2	M3	...
M1	0	14	7	...
M2	14	0	8	...
M3	7	8	0	...
...

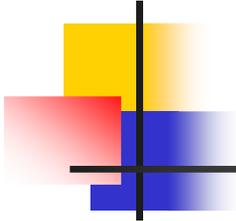


Lien



$$\forall i,j,k \quad \text{distance}(i,j) \stackrel{?, \leq}{=} \text{distance}(i,k) + \text{distance}(k,j)$$

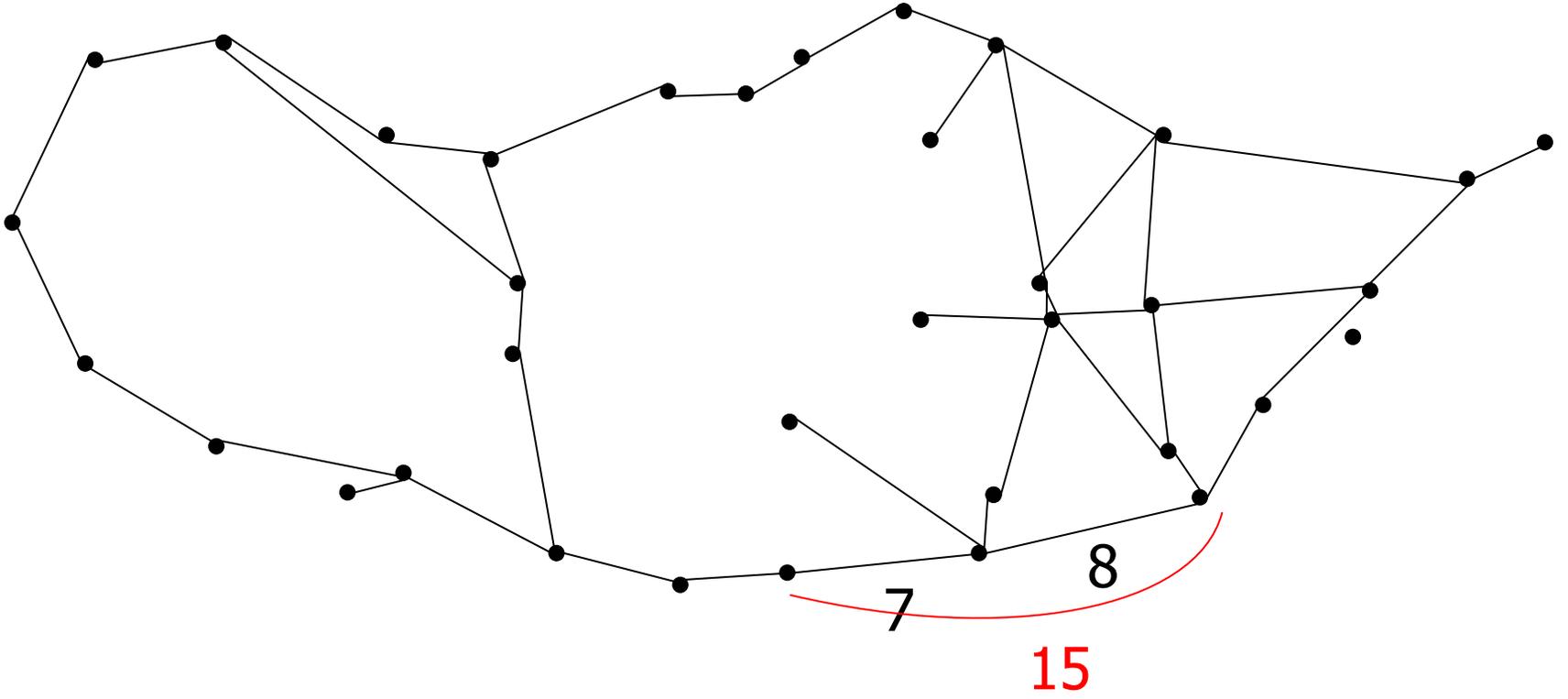
Vraisemblance multi-points (avec manquants) \Leftrightarrow
la distance entre deux points dépend du tour

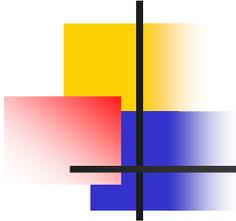


Problème du voyageur de commerce

Traveling Salesman Problem (TSP)

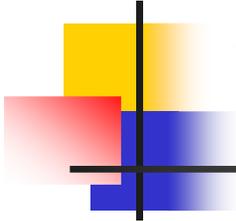
- Graphe complet
- Poids positifs sur chaque arête
 - Cas symétrique $\text{dist}(i,j) = \text{dist}(j,i)$
 - Inégalité triangulaire $\text{dist}(i,j) \leq \text{dist}(i,k) + \text{dist}(k,j)$
 - Distance euclidienne
- Trouver le plus court cycle Hamiltonien





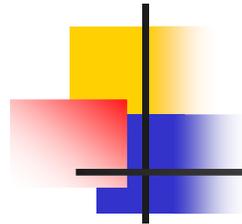
Problème du voyageur de commerce

- Intérêt théorique
 - Problème NP-complet
 - 1993-2001: +150 articles sur le TSP dans INFORMS & Decision Sciences
- Intérêt applicatif
 - Extension au problème de tournées de véhicules
 - Ordonnancement de marqueurs
 - Carthagène, NCBI/Concorde, ...



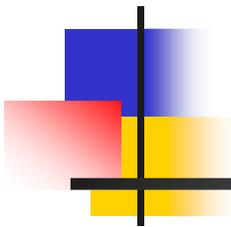
Variantes

- Euclidean Traveling Salesman Selection Problem
- Asymmetric Traveling Salesman Problem
- Symmetric Wandering Salesman Problem
- Selective Traveling Salesman Problem
- TSP with distances 1 and 2, TSP(1,2)
- K-template Traveling Salesman Problem
- Circulant Travelling Salesman Problem
- On-line Traveling Salesman Problem
- Time-dependent TSP
- The Angular-Metric Traveling Salesman Problem
- Maximum Latency TSP
- Minimum Latency Problem
- Max TSP
- Traveling Preacher Problem
- Bipartite TSP
- Remote TSP
- Precedence-Constrained TSP
- Exact TSP
- The Tour Cover problem
- ...

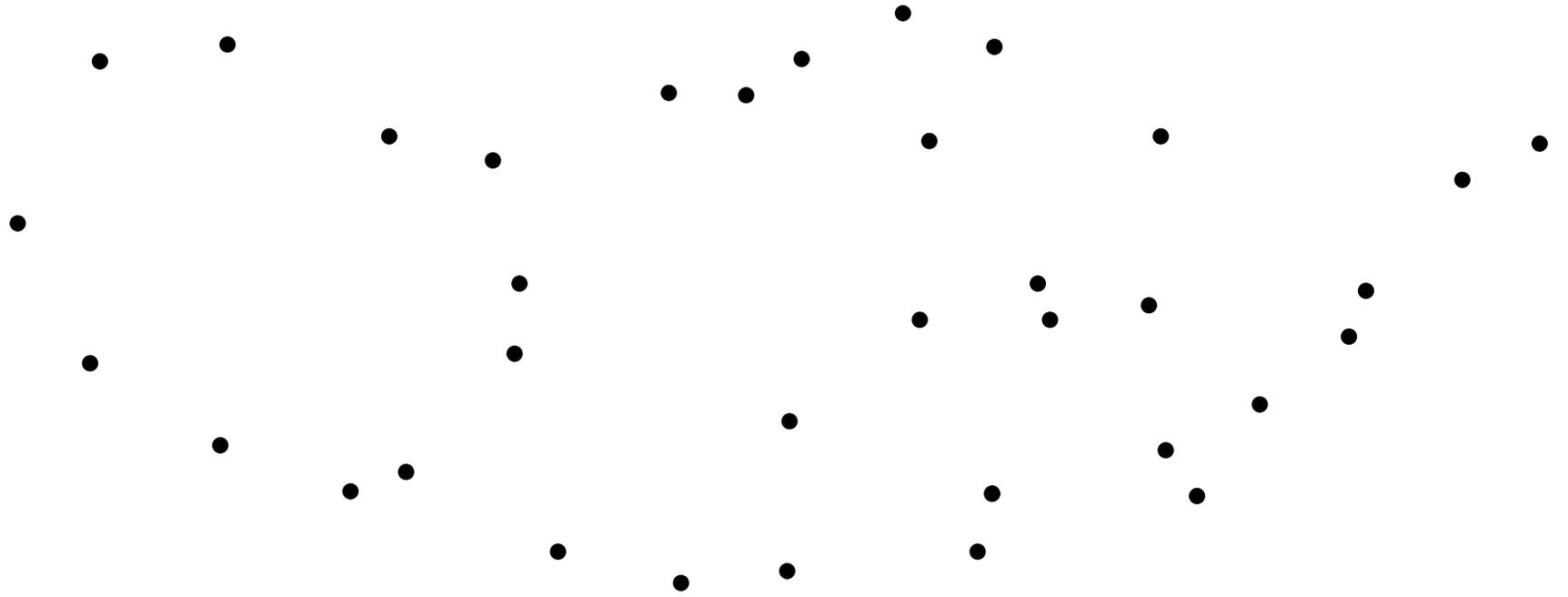


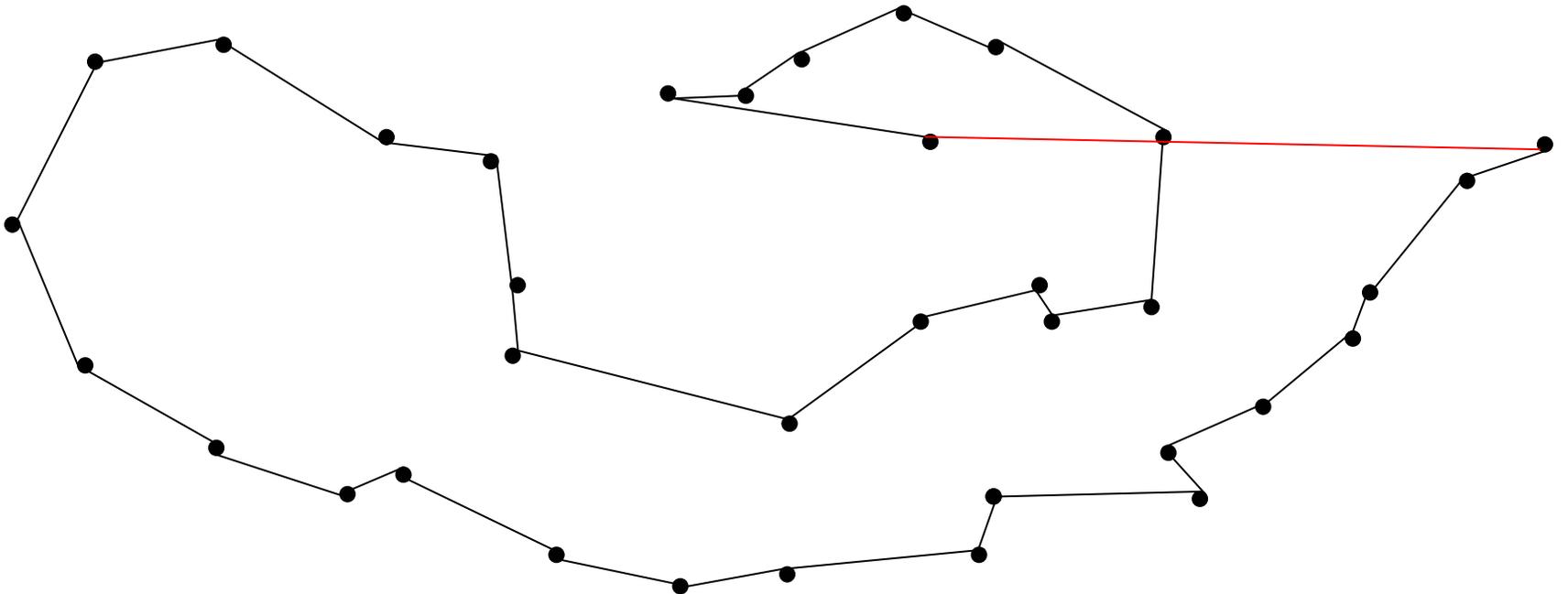
Plan

- Introduction du TSP
- Construction d'un tour
- Amélioration d'un tour
- Construction arborescente

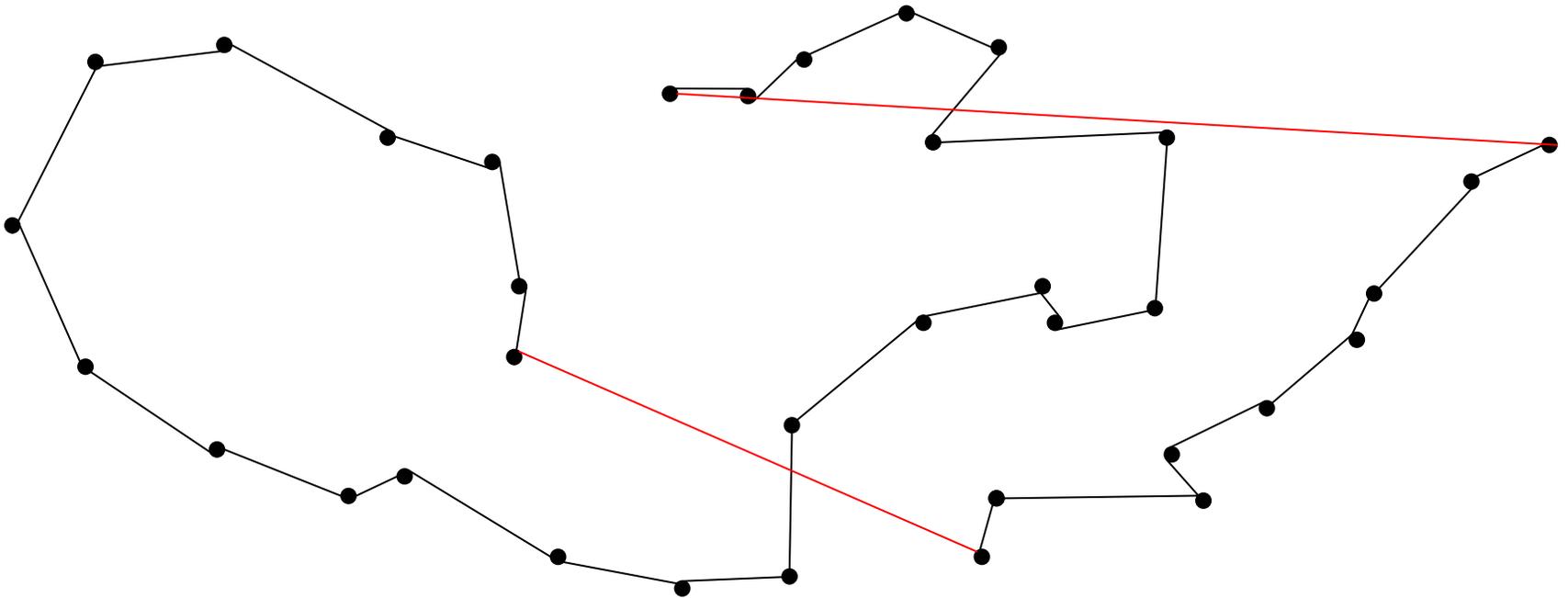


Construction d'un tour

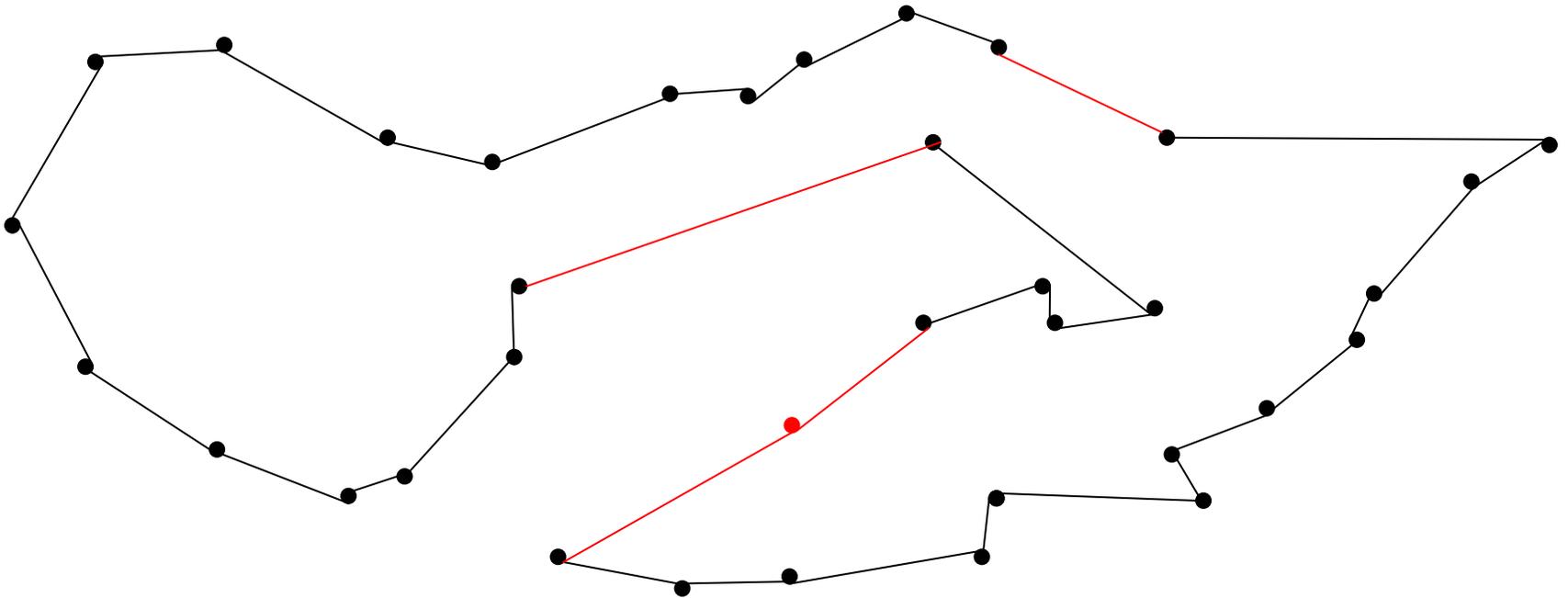




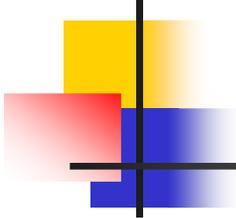
Heuristique « plus proche voisin »
(Nearest Neighbor)



Heuristique « plus petite distance 2-points »
(Greedy, multifragment heuristic)

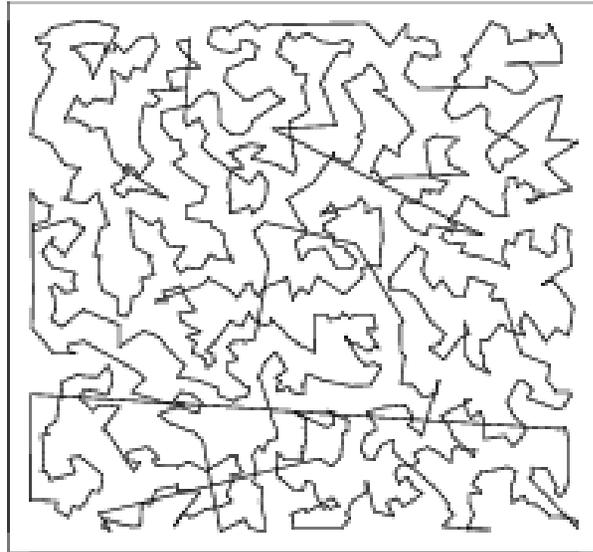


Heuristique « meilleure fusion »
(Savings, Clarke-Wright 1964)

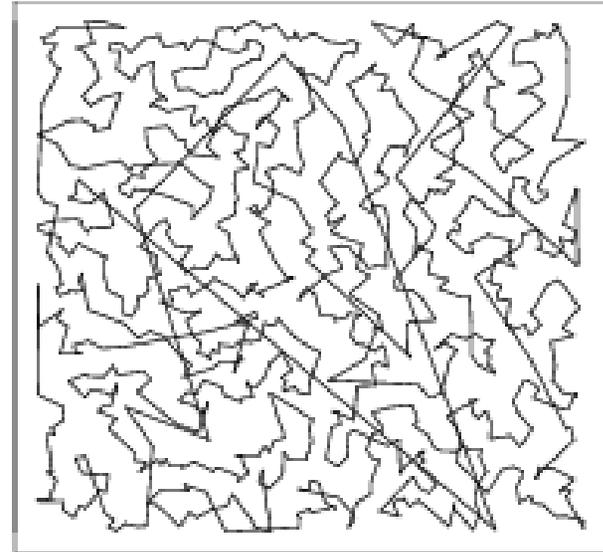


Heuristiques

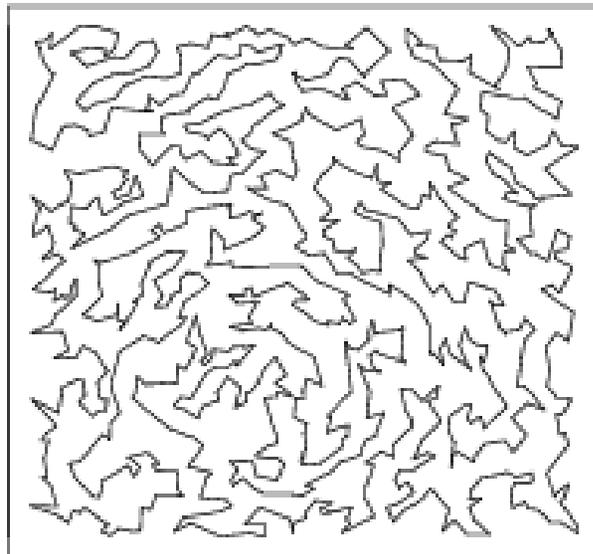
- Distance moyenne à l'optimum
 - Meilleure fusion : 11%
(Savings)
 - Plus petite distance 2-pts : 12%
(Greedy)
 - Plus proche voisin : 26%
(Nearest Neighbor)



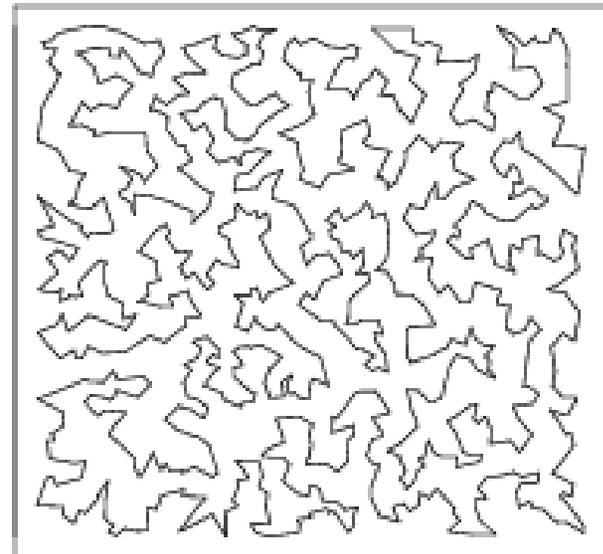
Greedy Tour



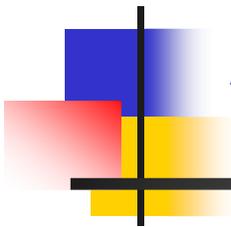
Nearest Neighbor Tour



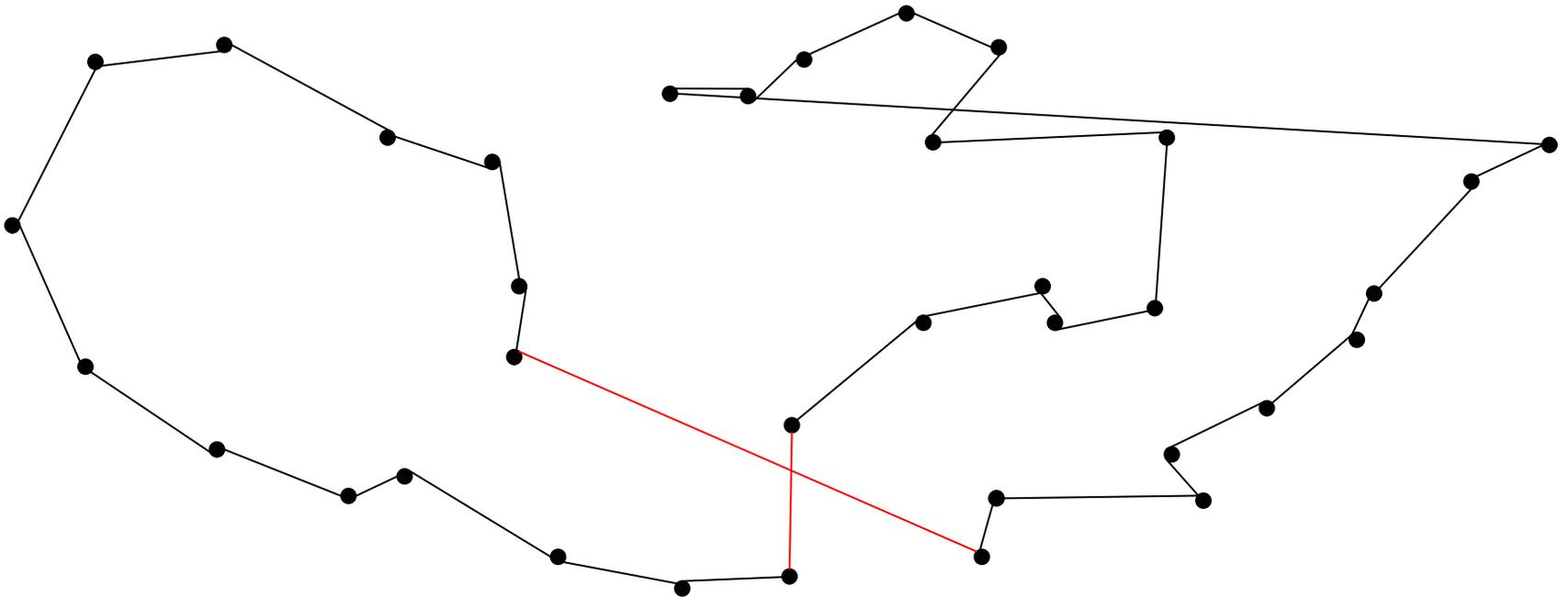
Savings Tour



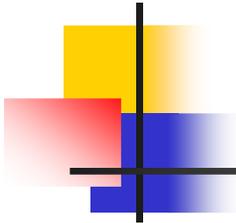
Optimal Tour



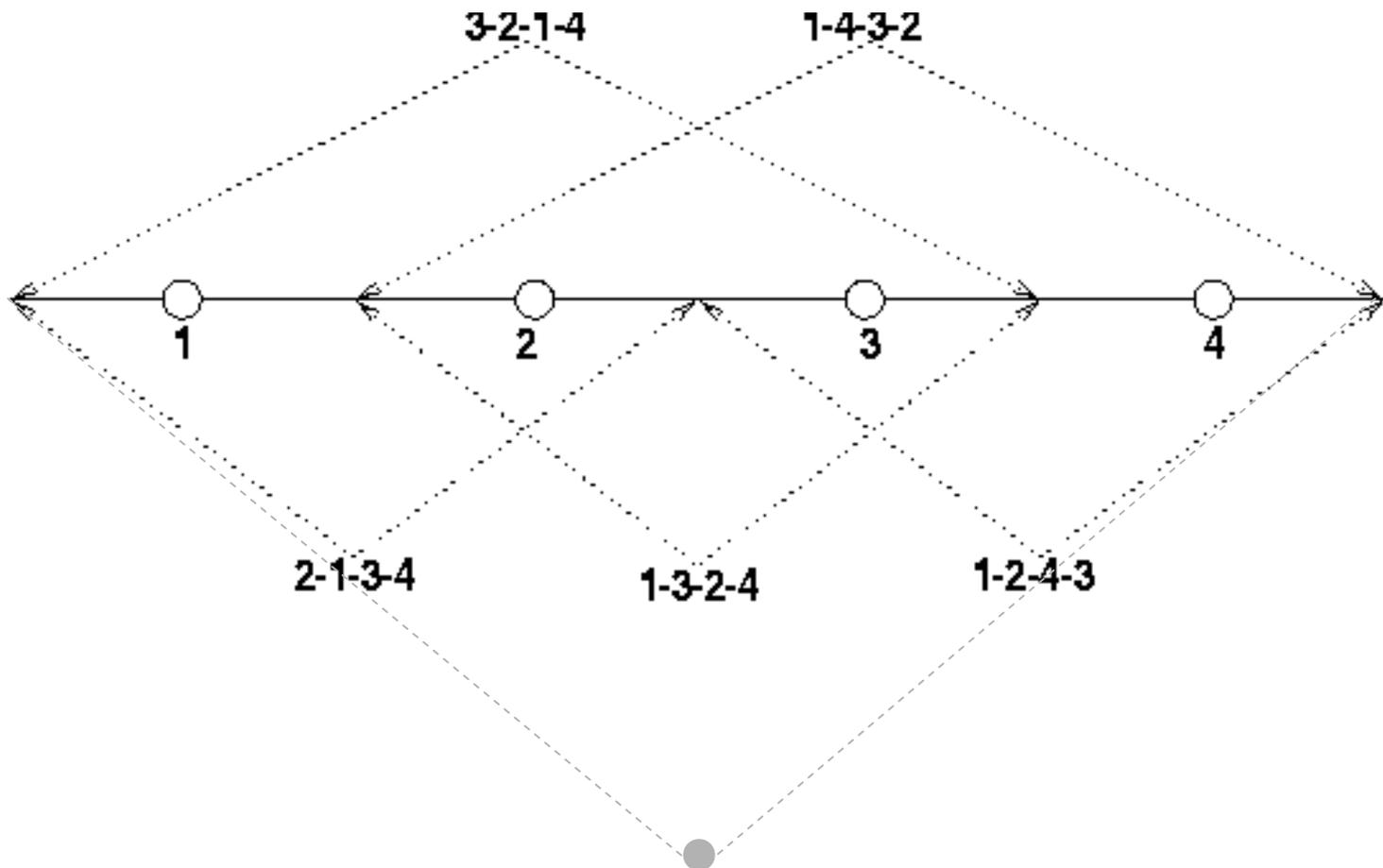
Amélioration d'un tour

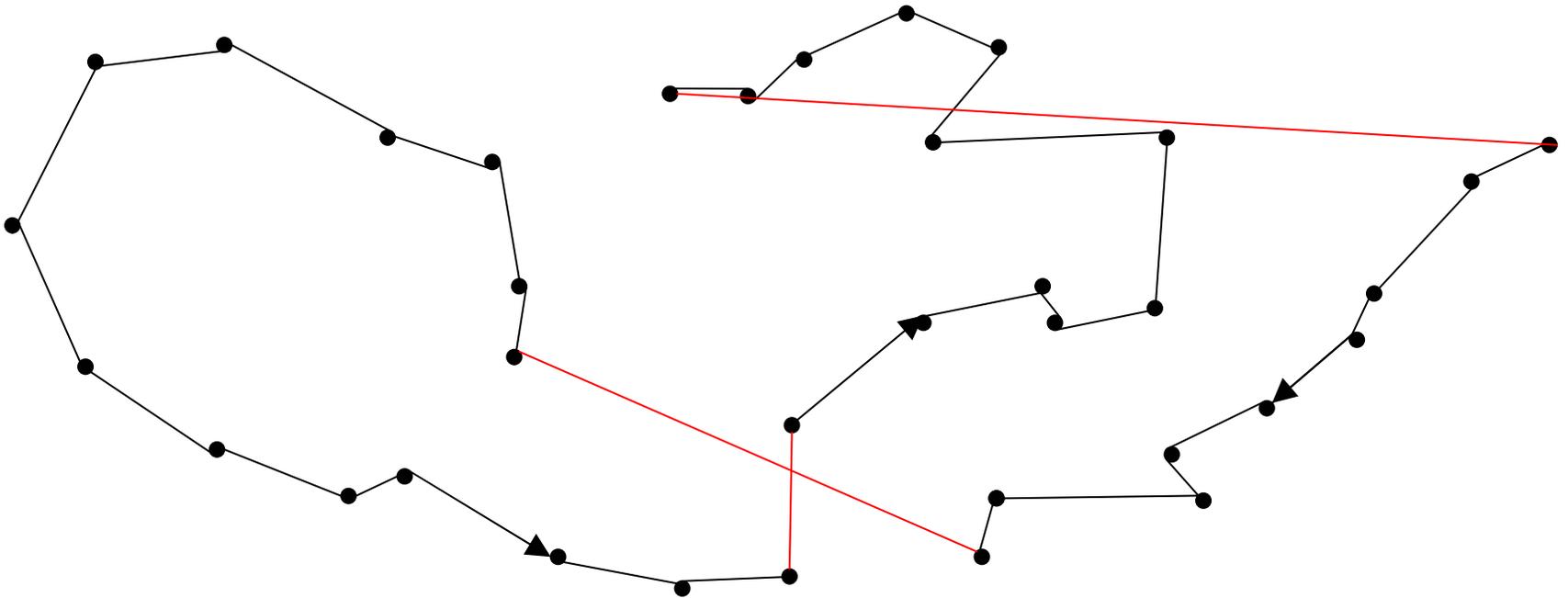


Effacer deux arêtes et reconstruire un tour
 \Leftrightarrow inversion d'une séquence de marqueurs



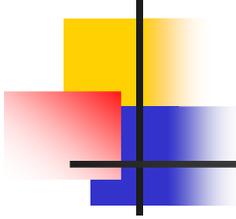
Modification *2-change*





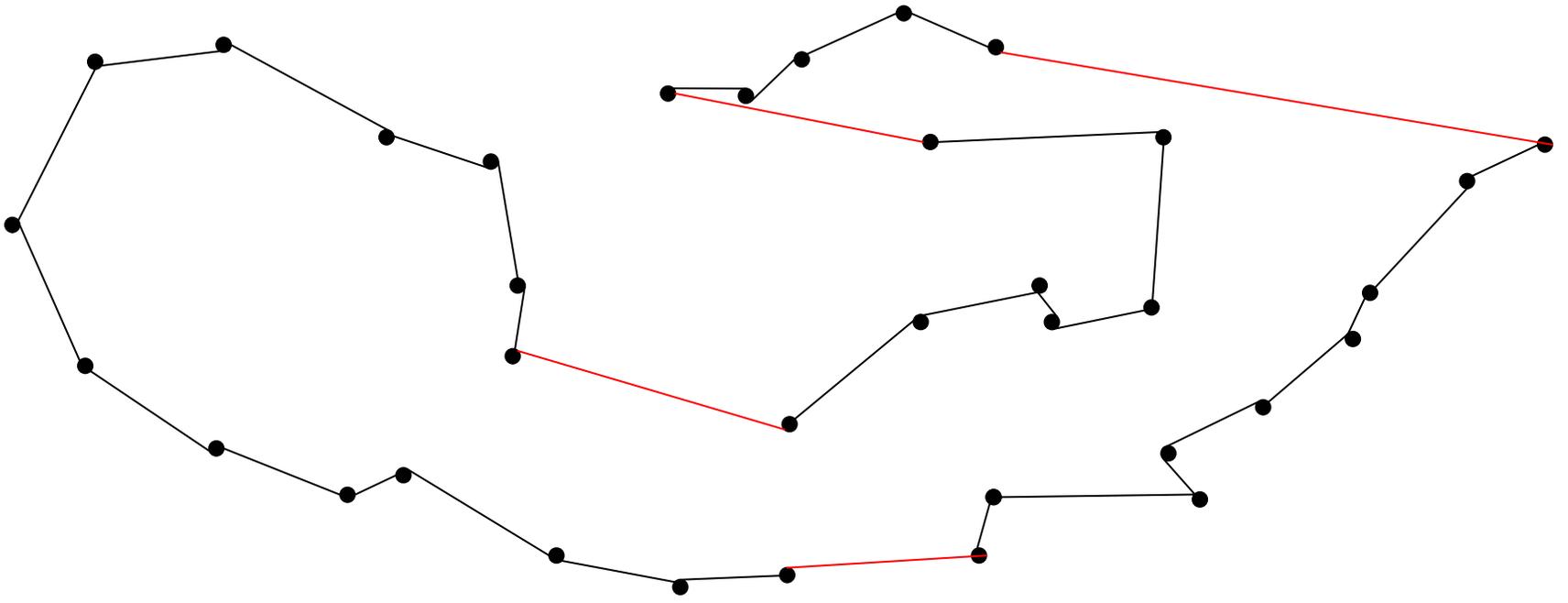
Effacer trois arêtes et reconstruire un tour (7 possibilités)

⇒ échange l'ordre de deux séquences de marqueurs

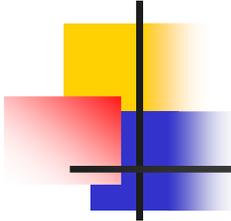


Recherche locale « gloutonne »

- 2-opt
 - Remarque : une succession finie de « 2-change » permet d'atteindre n'importe quel tour, y compris un tour optimum
 - Stratégie :
 - sélection du meilleur 2-change parmi $N*(N-1)/2$ voisins (2-move)
 - répétition tant que la distance totale diminue

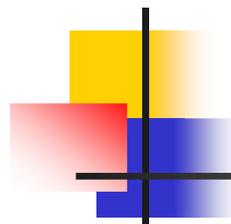


2-opt



Recherches locales gloutonnes

- Distance moyenne à l'optimum
 - 2-opt : 9%
 - 3-opt : 4%
 - LK (k-opt bridé) : 1%
- Complexité
 - 2-opt : $\sim N^3$
 - 3-opt : $\sim N^4$
 - LK (k-opt bridé) : $< N^4 ?$

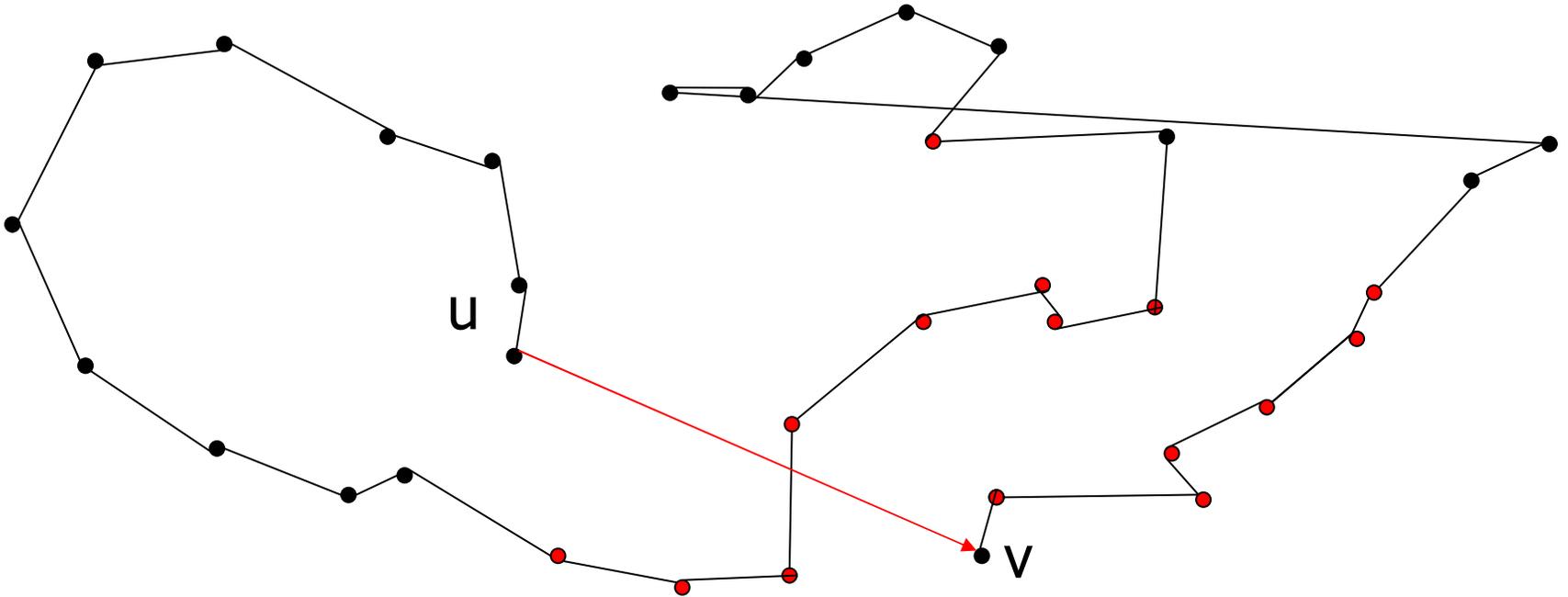


Complexités

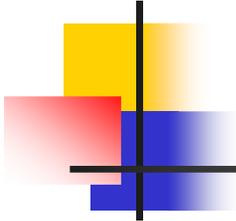
n = nombre de sommets

Opération	Complexité
A-TSP	$(n-1)!$
S-TSP	$(n-1)! / 2$
2-change	1
3-change	7
k-change	$(k-1)! \cdot 2^{k-1}$
k-move	$(k-1)! \cdot 2^{k-1} \cdot n! / (k! \cdot (n-k)!)$
	$\sim O(n^k)$ $k \ll n$
	en pratique : $o(n)$
2-opt et 3-opt	$\sim O(n^{k+1})$
	en pratique : $o(n^{1.2})$
	time(3-opt) $\sim 3 \times$ time(2-opt)

Implémentation efficace du 2-opt :



Pour chaque arête (uv) , maintenir la liste des sommets w tels que $\text{dist}(w,v) < \text{dist}(u,v)$



Lin & Kernighan (1973)

- k-change : $e_1 \rightarrow f_1, e_2 \rightarrow f_2, \dots$

$$\Rightarrow \sum_{i=1}^k (\text{dist}(e_i) - \text{dist}(f_i)) > 0$$

Il existe un ordre sur les i tel que toutes les sommes partielles soient positives :

$$S_l = \sum_{i=1}^l (\text{dist}(e_i) - \text{dist}(f_i)) > 0$$

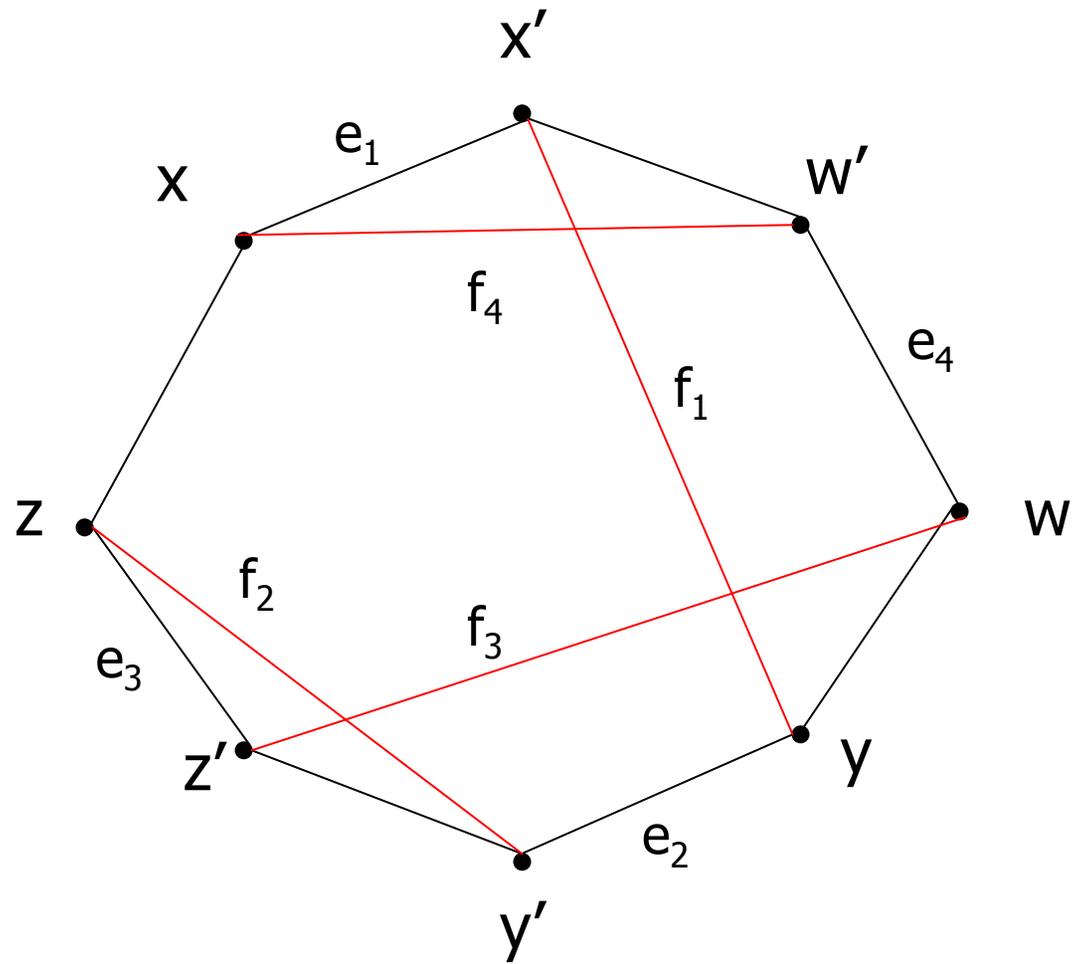
\Rightarrow Construit un cycle alterné valide augmentant :

$xx' \rightarrow yx', yy' \rightarrow zy', zz' \rightarrow wz', \text{ etc.}$

$\text{dist}(f_1) < \text{dist}(e_1), \text{dist}(f_1) + \text{dist}(f_2) < \text{dist}(e_1) + \text{dist}(e_2), \dots$

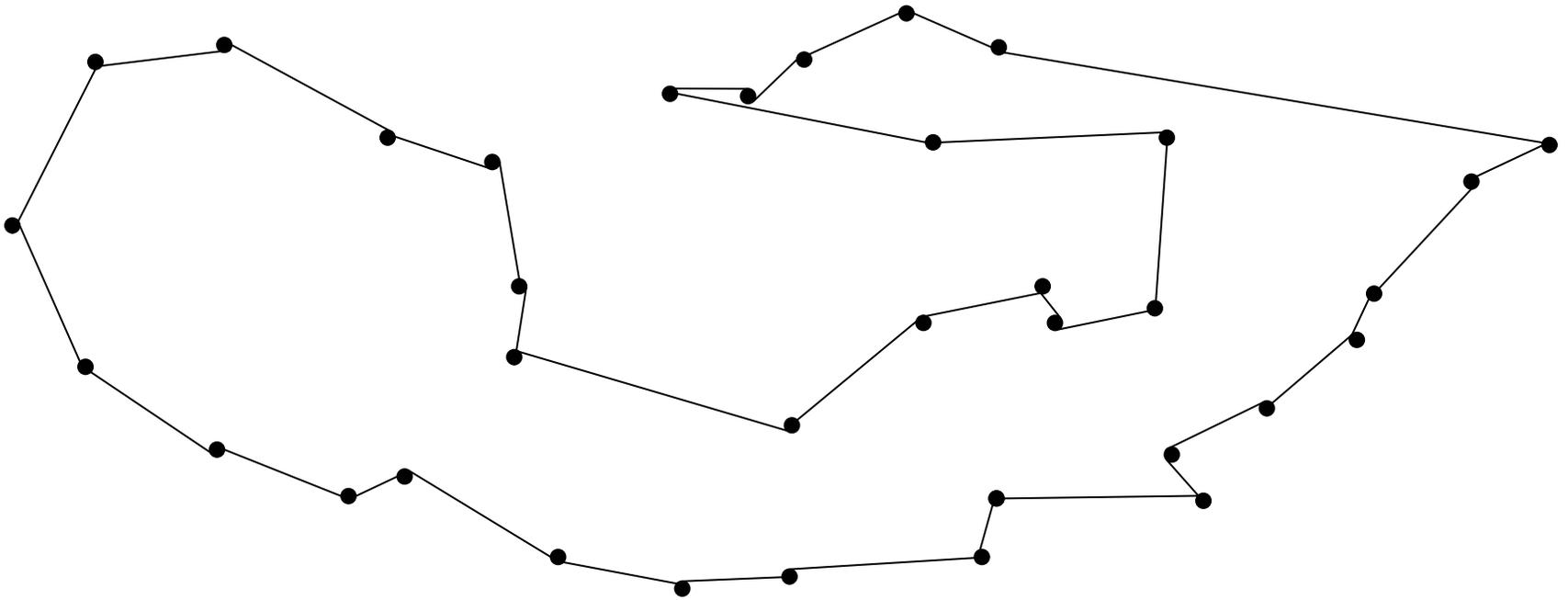
+ Backtrack sur le choix de y et z + Restart

(maximisation)

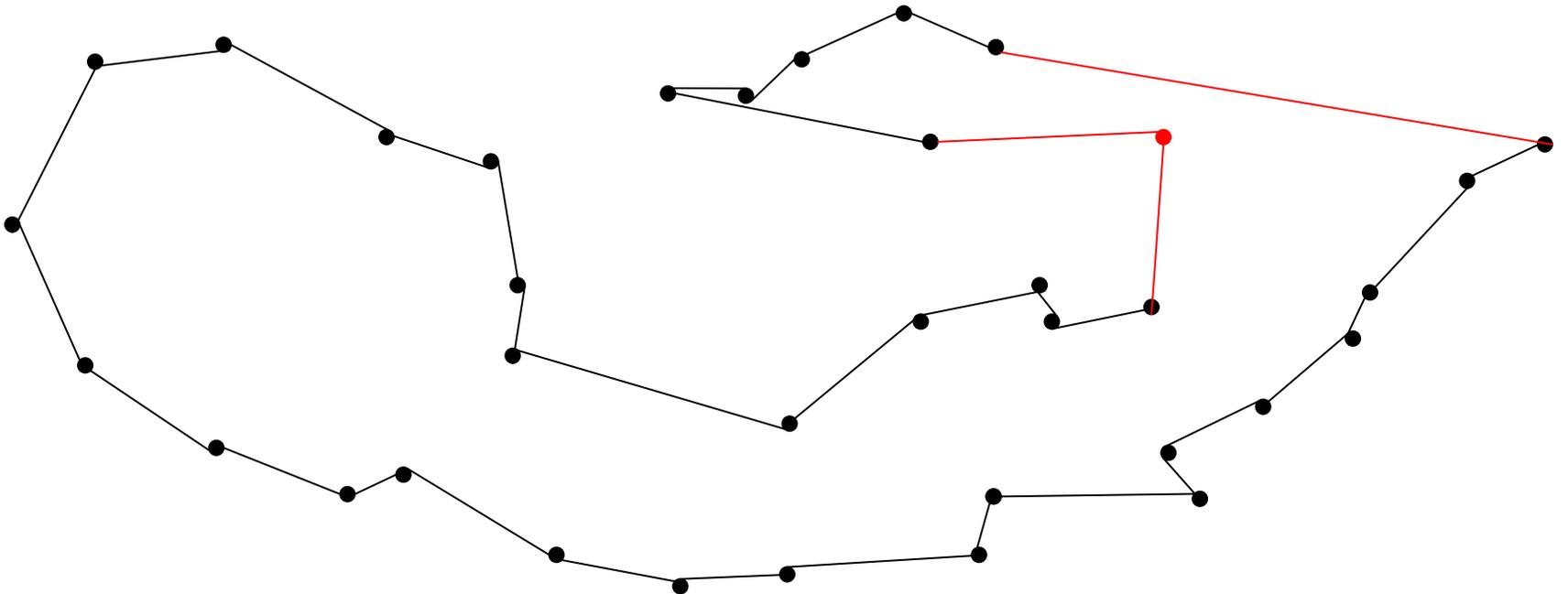


$$\{x, y, z, w, \dots\} \wedge \{x', y', z', w', \dots\} = 0$$

y parmi 5 meilleurs voisins de x', idem pour z et w.

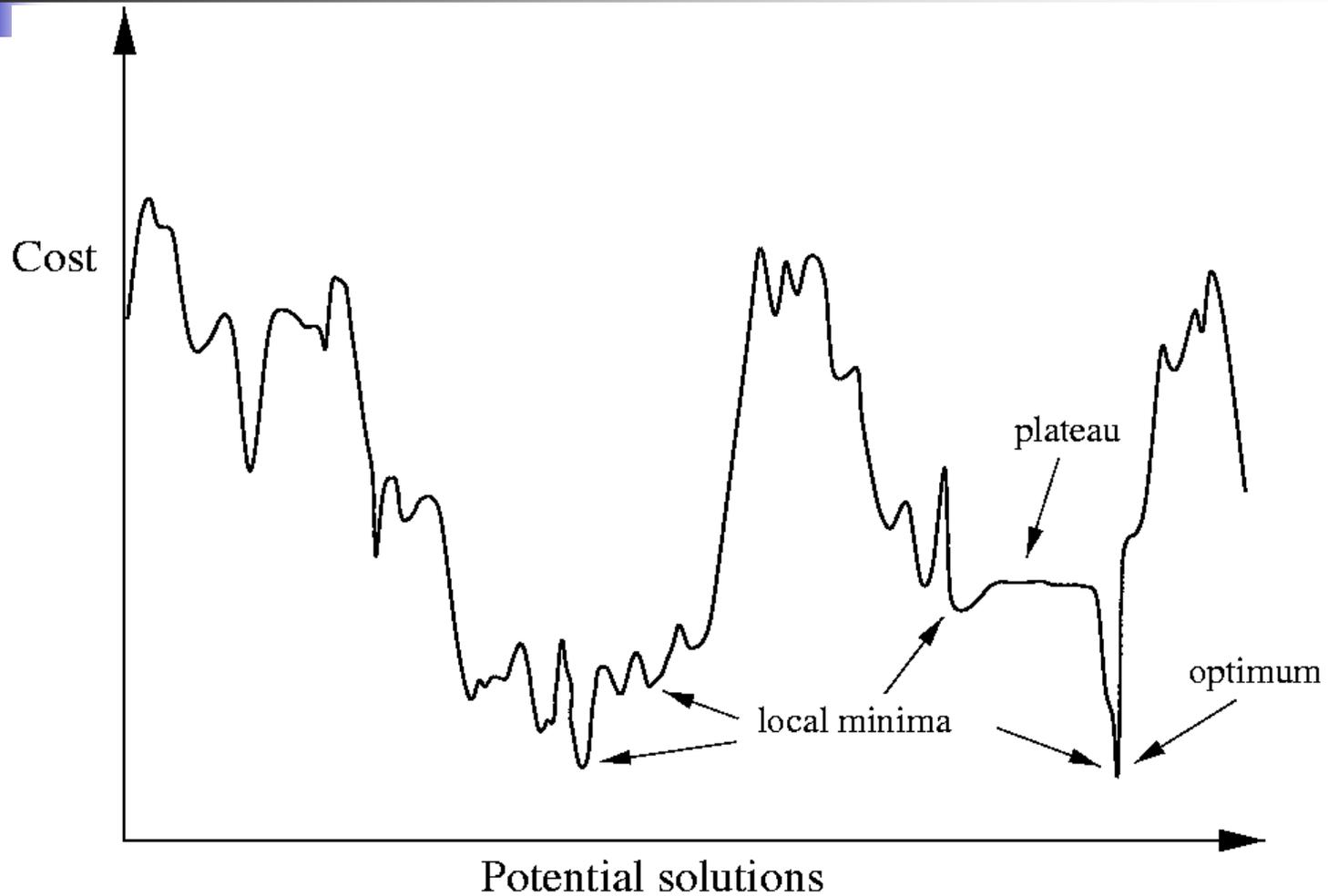


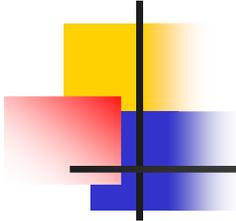
Est-ce que le tour 2-opt est un tour optimum ?



2-opt + réinsertion de sommet

Optimum *local* versus *global*



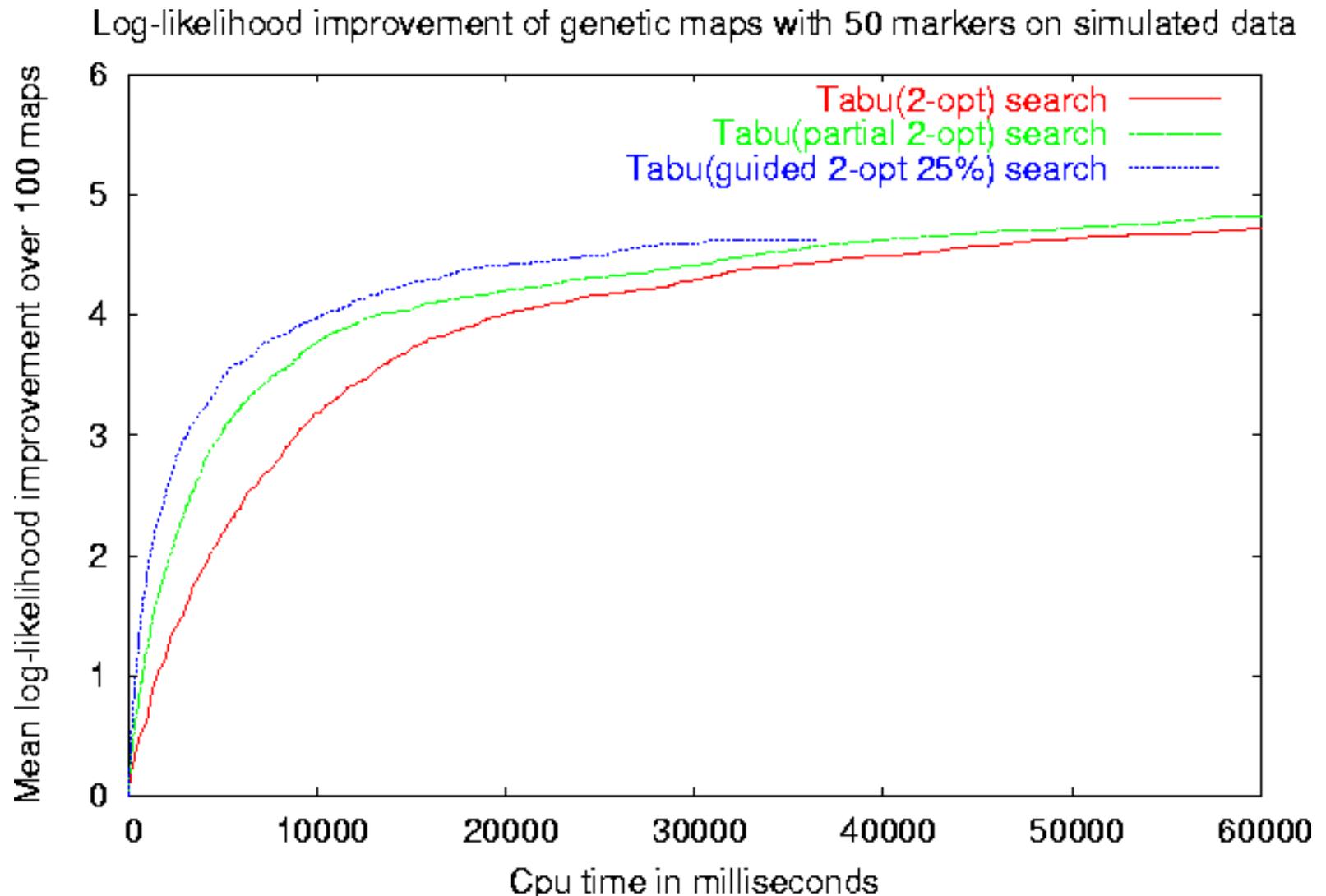


Recherches locales & « méta-heuristiques »

- Recherche Taboue
 - Sélection du meilleur voisin même s 'il dégrade la qualité du tour
 - Interdiction temporaire de refaire une modification locale faite précédemment
 - liste de modifications « taboues »
 - Ré-initialisation avec différents tours de départ
 - lorsque la recherche sélectionne un tour déjà vu
 - construction aléatoire d 'un tour

Expérimentations ds CartaGène

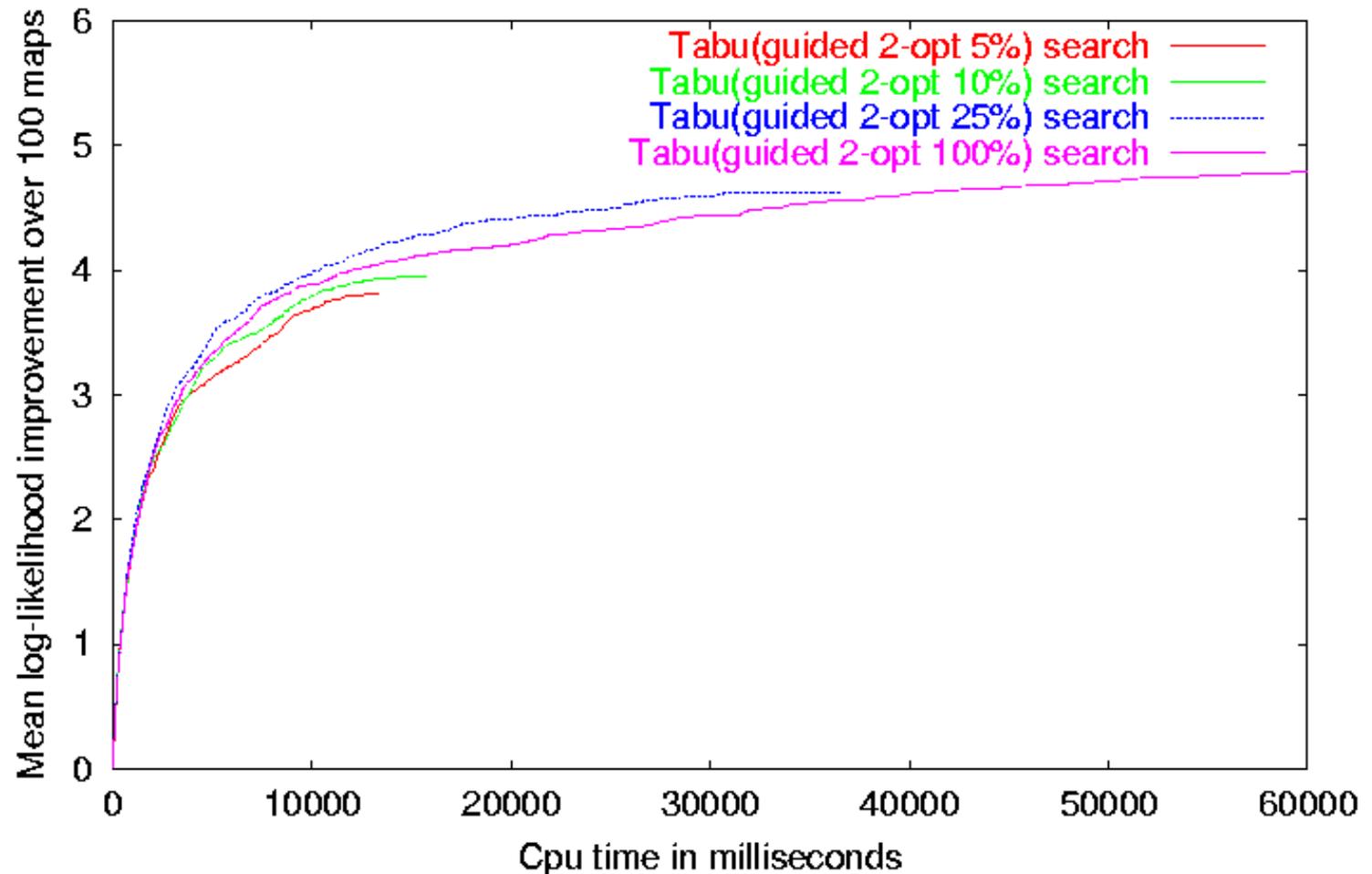
N=50 K=100 Err=30% Abs=30%

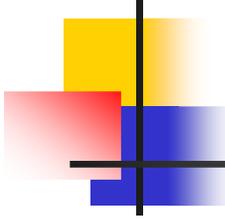


Légende : partial 2-opt = arrêt prématuré , guided 2-opt 25% = arrêt prématuré & trie avec X = 25 %

Expérimentations - suite

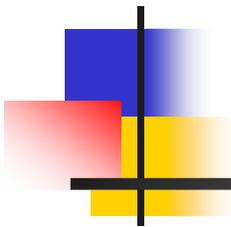
Log-likelihood improvement of genetic maps with 50 markers on simulated data





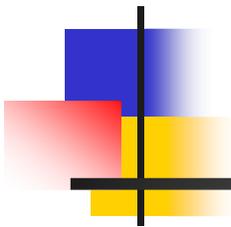
Autres méta-heuristiques

- Recuit simulé (*simulated annealing*)
 - Choix aléatoire des modifications locales
 - Acceptation d'un voisin en fonction de sa qualité ; processus de + en + glouton
- Algorithmes génétiques
 - Population d'individus (tours)
 - Mutation, croisement,...

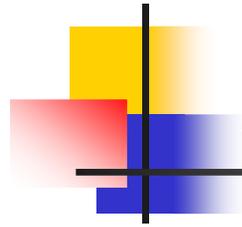


Recherche locale

Démonstration



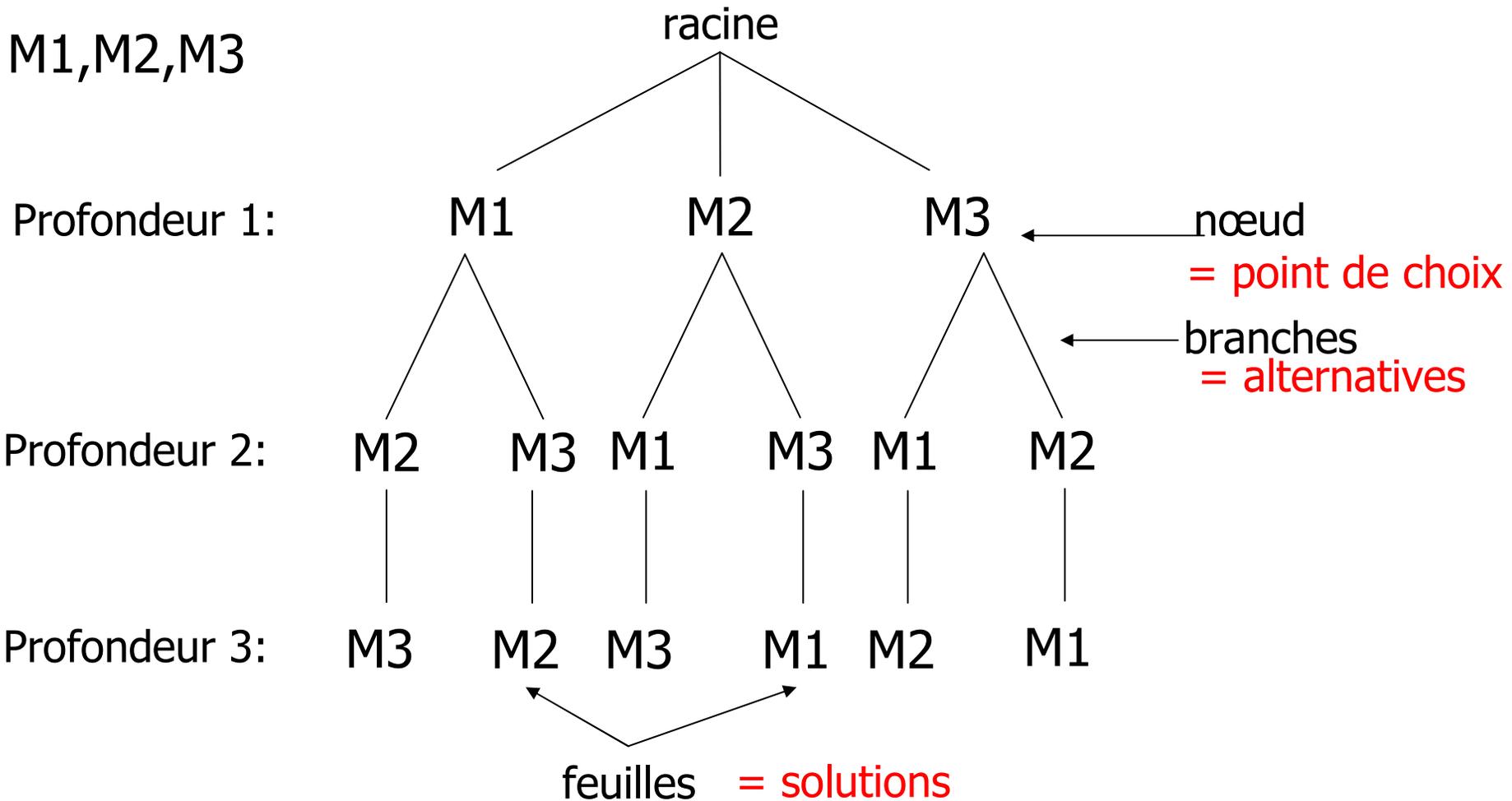
Construction arborescente

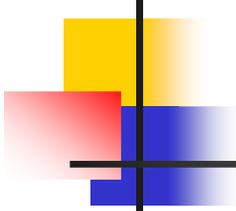


Arbre de recherche



M1,M2,M3



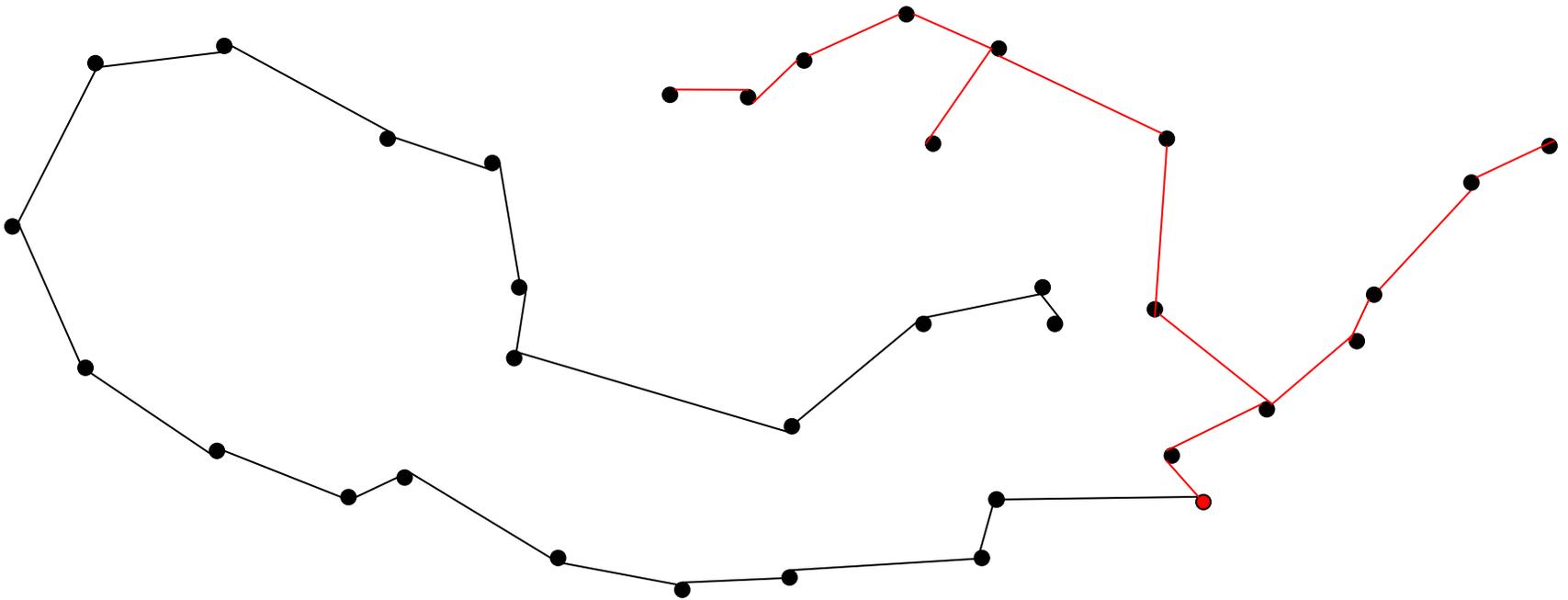


Recherche globale

- Complexité : $n!/2$ ordres différents
 - Éviter les ordres symétriques (1^{ère} moitié de l'arbre)
- Utilisation des heuristiques gloutonnes pour ordonner les alternatives
- Algorithme de « séparation & évaluation »
 - Élagage des branches qui ne conduisent pas vers une solution meilleure
- Possibilité de combiner recherche locale et recherche globale

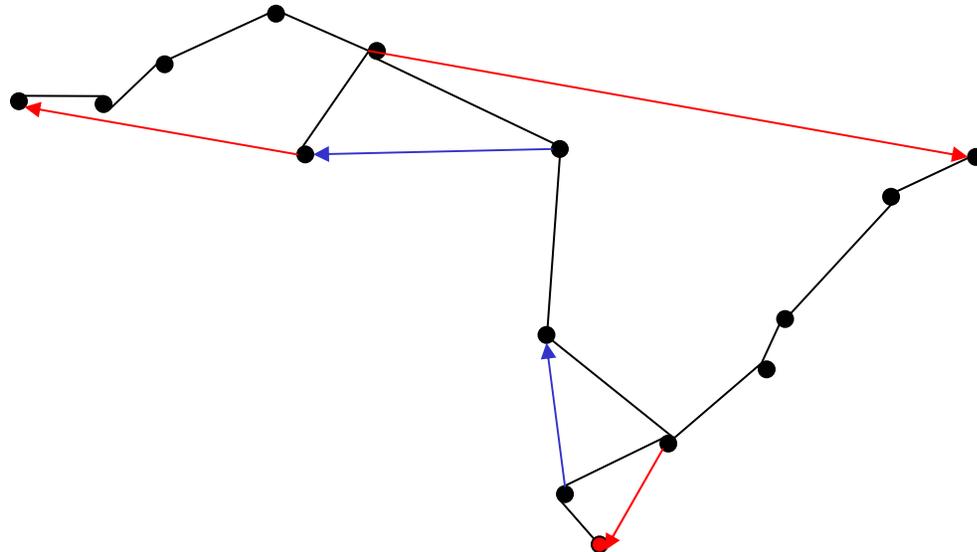
Elagage de branche

Arbre couvrant de poids minimum



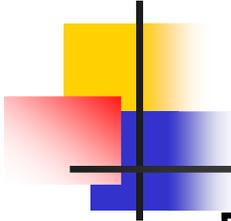
Algorithme de Prim (1957)

Algorithme de Held & Karp (arbre plus « filiforme ») (1971)
 \leq linear programming relaxation of TSP, $LB(I)/OPT(I) \geq 2/3$



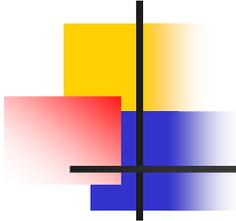
Heuristique de Christofides (1976)

$\Rightarrow A(I) / \text{OPT}(I) \leq 3/2$ (avec inégalité triangulaire)



Complexité

Complexité	Ordinateur de référence	Ordinateur 100 fois plus rapide	Ordinateur 1000 fois plus rapide
N	$N1$	$100 * N1$	$1000 * N1$
N^2	$N2$	$10 * N2$	$31,6 * N2$
N^3	$N3$	$4,64 * N3$	$10 * N3$
2^N	$N4$	$N4 + 6,64$	$N4 + 9,97$
3^N	$N5$	$N5 + 4,19$	$N5 + 6,29$



Méthodes complètes

- 1954 : 49 villes
- 1971 : 64 villes
- 1975 : 100 villes
- 1977 : 120 villes
- 1980 : 318 villes
- 1987 : 2,392 villes
- 1994 : 7.397 villes
- 1998 : 13.509 villes
- 2001 : 15.112 villes (585936700 sec. \approx 19 ans CPU)