Hybrid Local Search Methods for Discrete Optimization Problems

Yury Kochetov

Sobolev Institute of Mathematics
Novosibirsk State University

XIII International Asia School-seminar
Novosibirsk 2017

Outline

- Local search and metaheuristics
  - Simulated Annealing
  - Great Deluge
  - Tabu Search
  - Variable Neighborhood Search
  - Genetic Algorithms

- Hybrid methods with mathematical programming tools
  - Relaxation Induced Neighborhood Search
  - Local Branching
  - Large Neighborhoods
  - Core Concepts
  - Feasibility Pump

Standard local descent algorithm

1. Select a starting solution $s \in Sol$
2. Try to find the best neighboring solution s':

$$f(s') < f(s), \qquad s' \in N(s)$$

3. If it exists then move:

s := s' and go to 2 else STOP

4. Return local minimum s.

Examples: – Simplex method for linear programming problem;

– Ford–Fulkerson method for maximum flow problem;

– Bubble sort algorithm for the sorting problem.

Advanced Local Search Strategies

Optimization problem $\min\{F(s) \mid s \in Sol\}$

Neighborhood $N: Sol \to 2^{Sol}$

Threshold algorithms

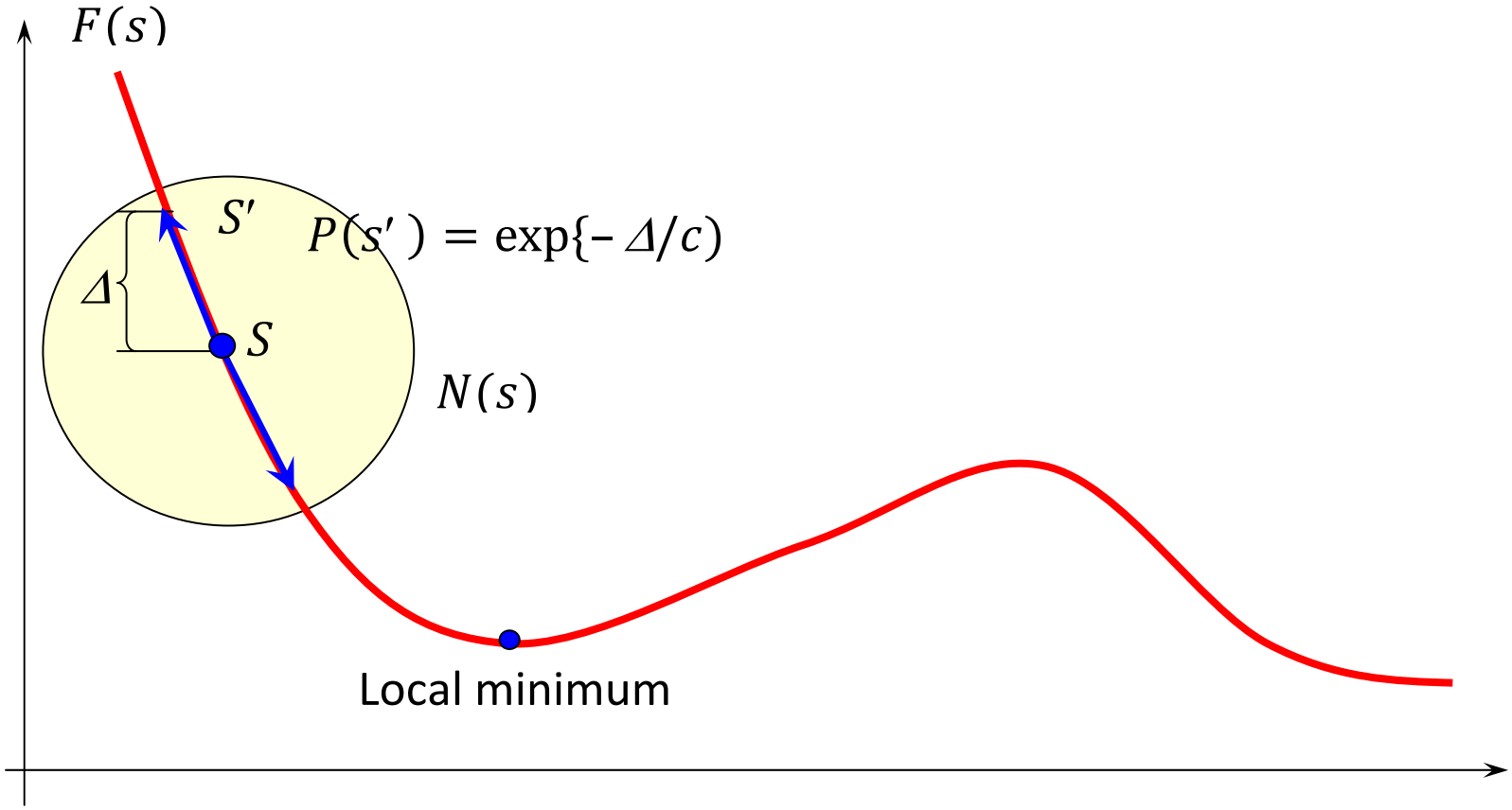
1. Construct an initial solution $s \in Sol$, $s^* := s$, $k := 0$.
2. Repeat until a stop criterion is satisfied
   - 2.1. Generate $s' \in N(s)$ and put $k := k + 1$;
   - 2.2. If $F(s) - F(s') < t_k$ then $s := s'$;
   - 2.3. If $F(s^*) > F(s)$ then $s^* := s$.
3. Return s^*.

Three Types of Threshold Algorithms

- Iterative improvement: $t_k = 0, k \geq 0$, it is standard LD algorithm with random pivoting rule.
- Threshold accepting: $t_k \geq 0,\ t_k \geq t_{k+1},\ k \geq 0,\ \lim_{k\to\infty} t_k = 0$ ..
- Simulated annealing: t_k is a random variable with expected value $E(t_k) = c_k, k \geq 0$; more exactly, the probability of accepting $s' \in N(S)$ at the k^{th} iteration is given by

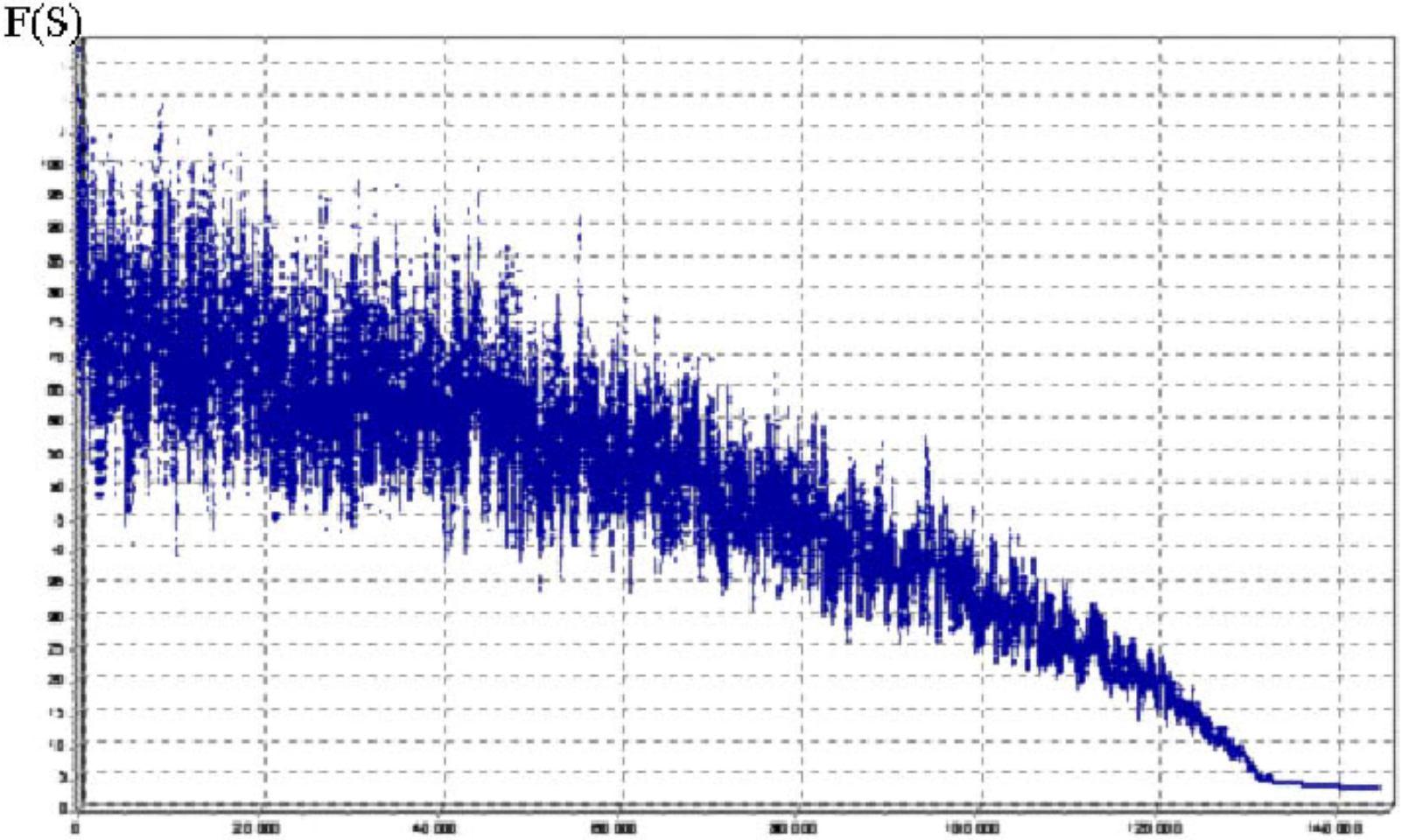
$$P_{c_k}\{\text{accept } s'\} = \begin{cases} 1 & \text{if } F(S') \leq F(S) \\ exp\left(\dfrac{F(S) - F(S')}{c_k}\right) & \text{if } F(S') > F(S) \end{cases}$$

Simulated Annealing Algorithm



SA was introduced by Kirkpatrick, Gellat, and Vecchi, 1983; Černỳ, 1985.

Typical Behavior of SA



Theorem. Let $(\Im, f)$ be an instance of a combinatorial optimization problem, $\mathcal{N}$ a neighborhood function, and $P(k)$ the transition matrix of the homogeneous Markov chain associated with the simulated annealing algorithm with $c_k = c$ for all k. Furthermore, let the following condition be satisfied:

$$\forall i, j \in \Im \;\; \exists p \geq 1 \;\; \exists l_0, l_1, \ldots, l_p \in \Im$$

with

$$l_0 = i, \;\; l_p = j, \;\; \text{and} \;\; G_{l_k l_{k+1}} > 0, \quad k = 0,1, \ldots, p-1$$

Then the associated homogeneous Markov chain has a stationary distribution $q(c)$, whose components are given by

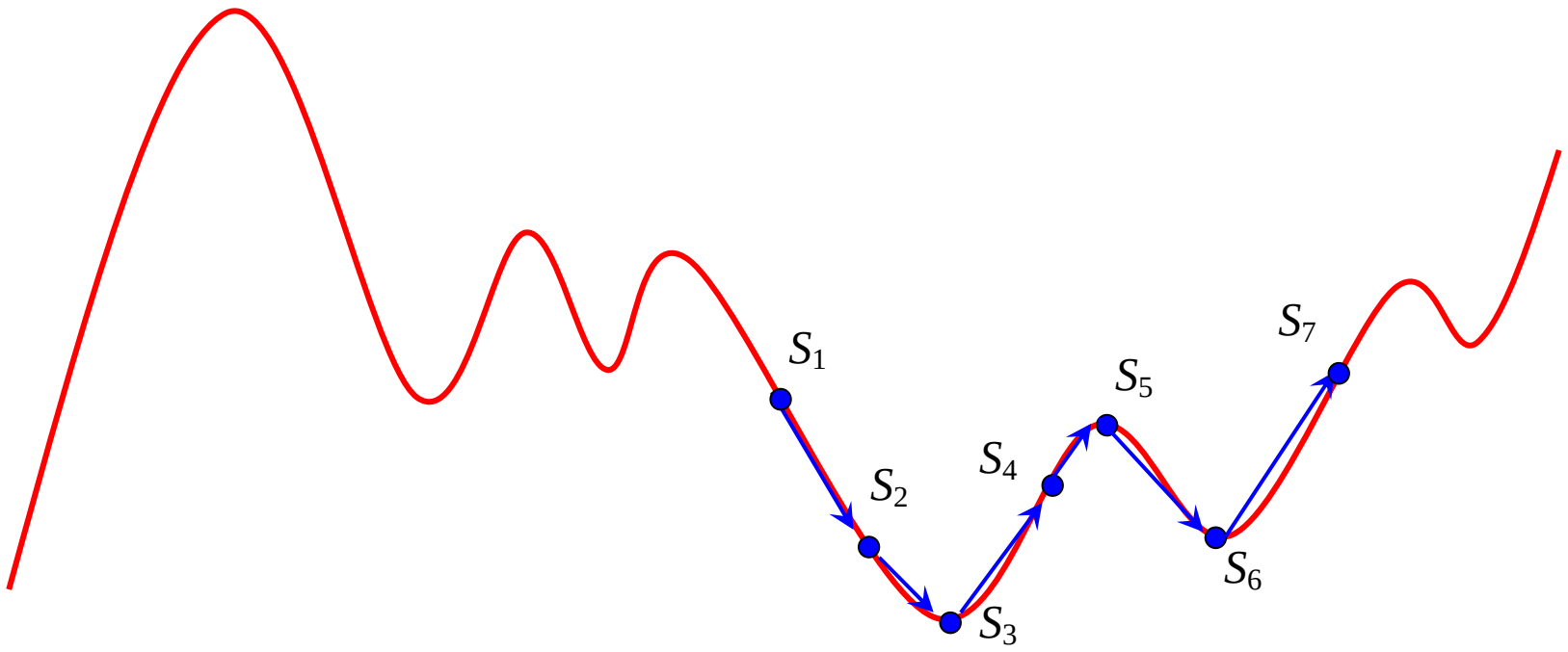
$$q_i(c) = \frac{\exp\left(-f(i)/c\right)}{\sum_{j \in \Im} \exp\left(-f(j)/c\right)} \quad \text{for all} \;\; i \in \Im$$

and

$$q_i^* \stackrel{\text{def}}{=} \lim_{c \downarrow 0} q_i(c) = \frac{1}{|\Im^*|} \chi_{(\Im^*)}(i)$$

with i^* denotes an optimal solution, and $\Im^*$ the set of optimal solutions.

Tabu Search Algorithm



Tabu Search was introduced by F. Glover 1989

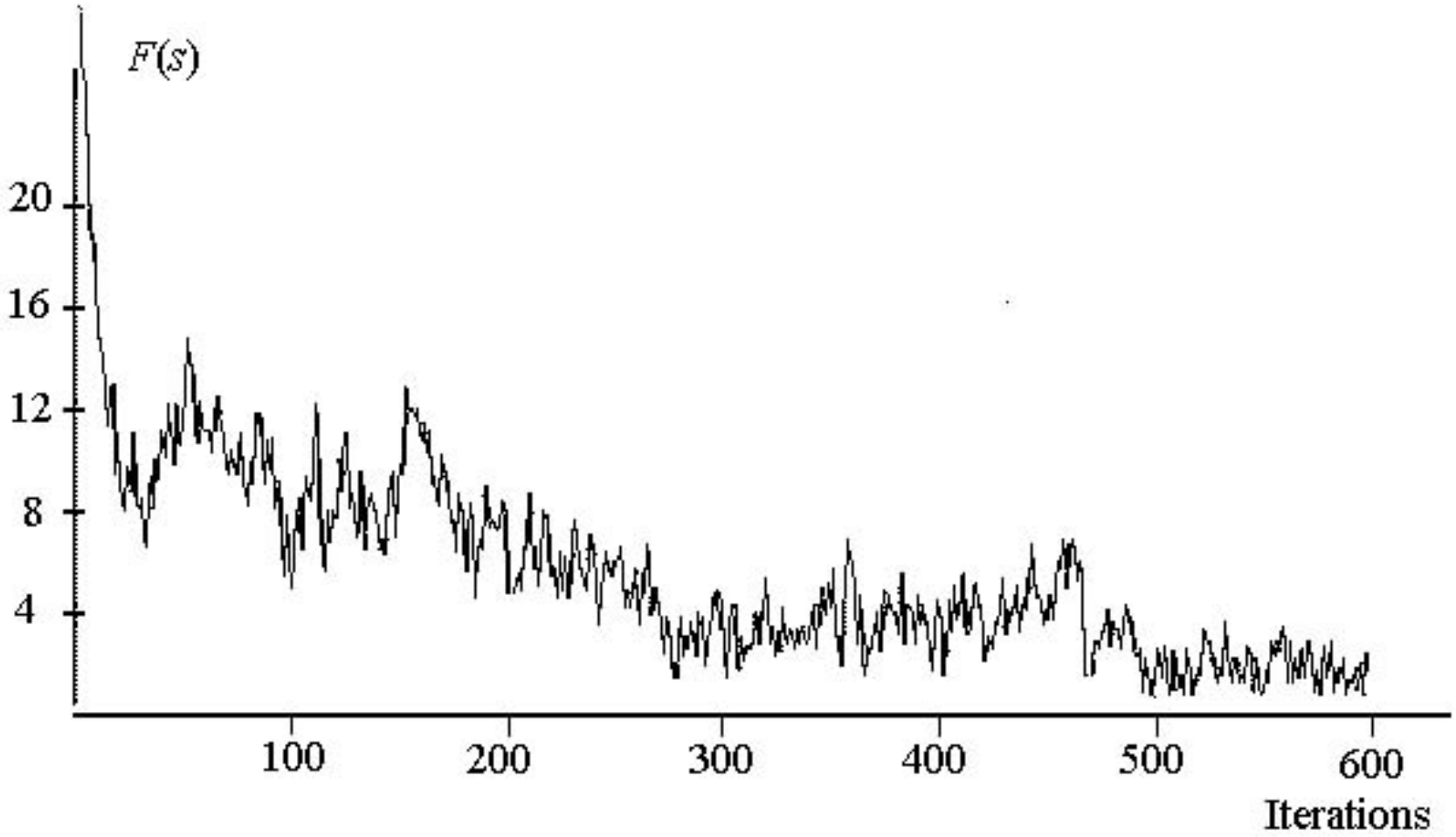
Probabilistic Tabu Search

1. Construct an initial solution $s \in Sol$; $s^* := s$;
2. Repeat until a stop criterion is satisfied
   2.1. Generate a random subneighborhood $N'(s) \subseteq N(s)$
   2.2. Select the best legal neighbor s' :

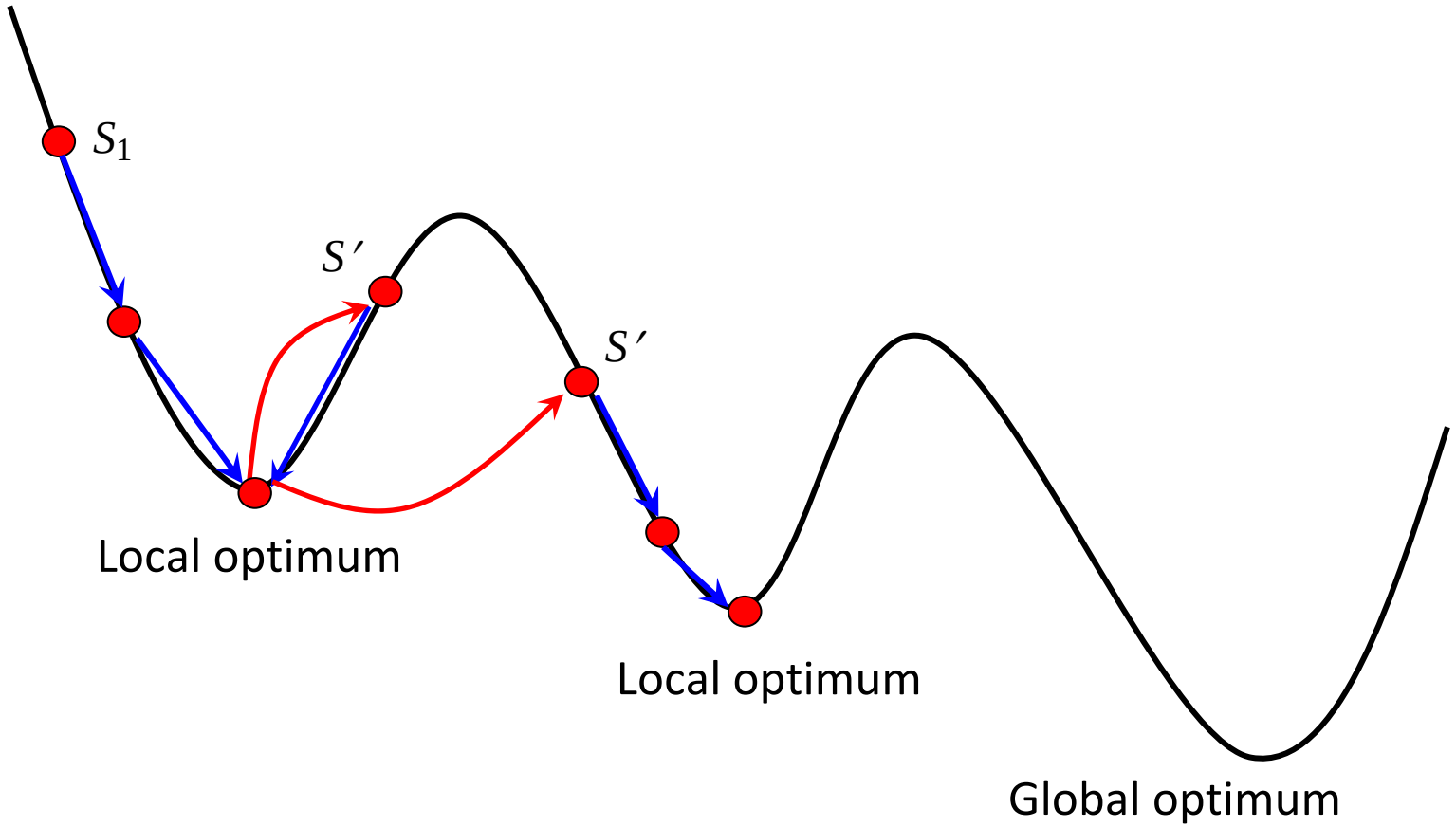
$$s' : F(s') = \min\{F(s'') \mid s'' \in N'(s) \backslash \text{Tabu}(s)\}$$

   2.3. Put $s := s'$, update TabuList
   2.4. If $F(s^*) > F(s)$ then $s^* := s$.
3. Return s^*.

Typical Behavior of Tabu Search



Variable Neighborhood Search Algorithm

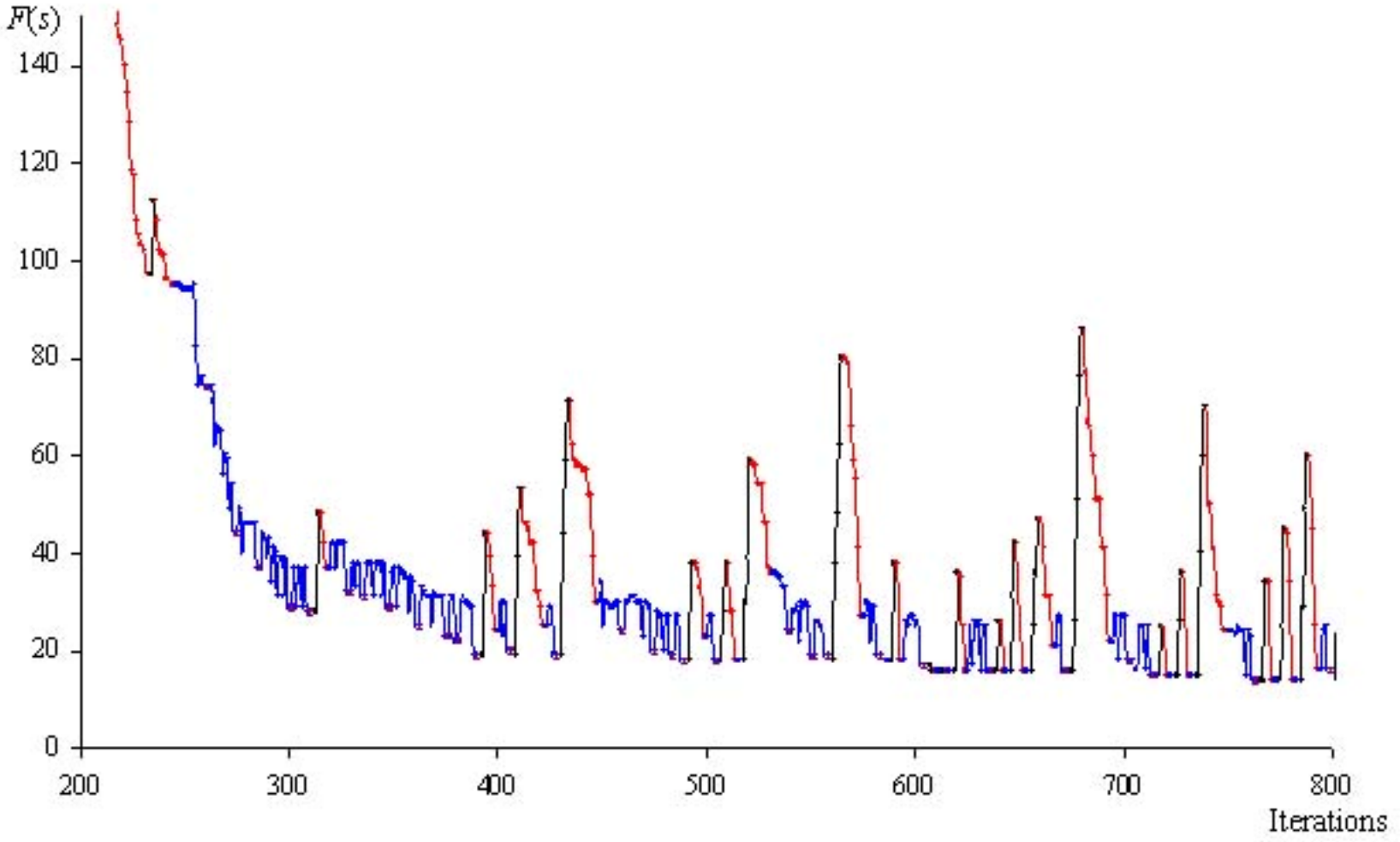


VNS was introduced by P. Hansen and N. Mladenović 1997.

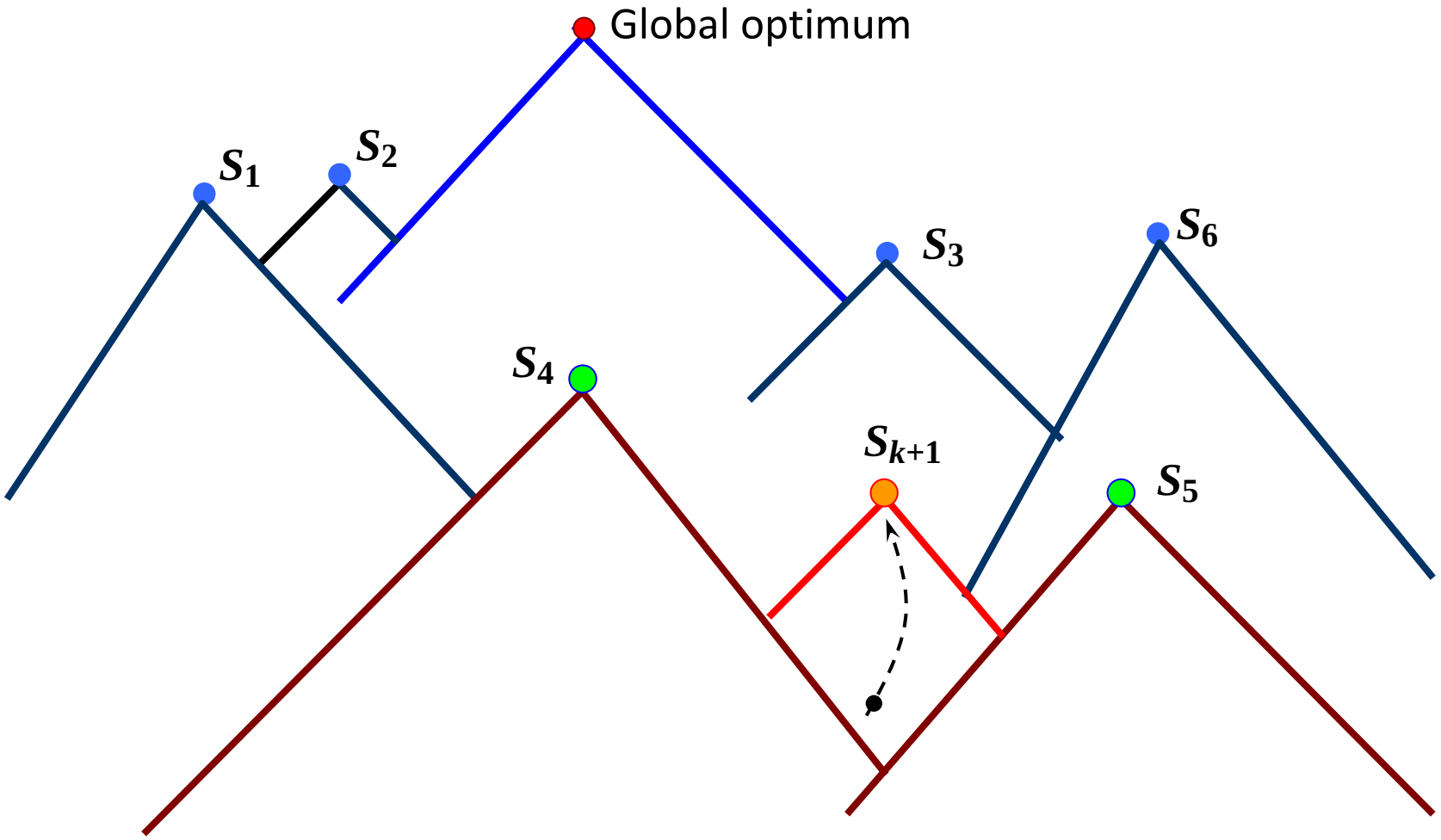
VNS Algorithm

1. Construct an initial solution $s \in Sol$, select the set of neighborhoods N_k, $k = 1, \ldots, k_{\max}$
2. Repeat until a stop criterion is satisfied
    2.1. Set $k := 1$;
    2.2. Repeat until $k = k_{\max}$;
        (a) Generate $s' \in N_k(s)$ at random;
        (b) Apply some local search method with s' as initial solution; denote with s'' the so obtained local minimum;
        (c) $If\ F(s'') < F(s)$ then $(s := s'')\&(k := 1)$ else $k := k + 1$
3. Return s.

Typical Behavior of Variable Neighborhood Search



Genetic Local Search Algorithm



Genetic algorithms approach was introduced by J.H. Holland, 1975.

GLS Algorithm

1. Construct an initial population of k solutions.
2. Use local search to replace the solutions in the population by local optima.
3. Repeat until a stop criterion is satisfied
   1.1. Augment the population by adding m offspring solutions;
   1.2. Use local search to replace the m offspring solutions by m local optima;
   1.3. Reduce the population to its original size by selecting k solutions from the current population.
4. Return the best solution from the population.

Local branching

Given a solution $\tilde{y}$ and integer positive k. We consider the mixed integer problem:

$$\min \{cx + fy\}$$

$$Ax + By \geq b;$$

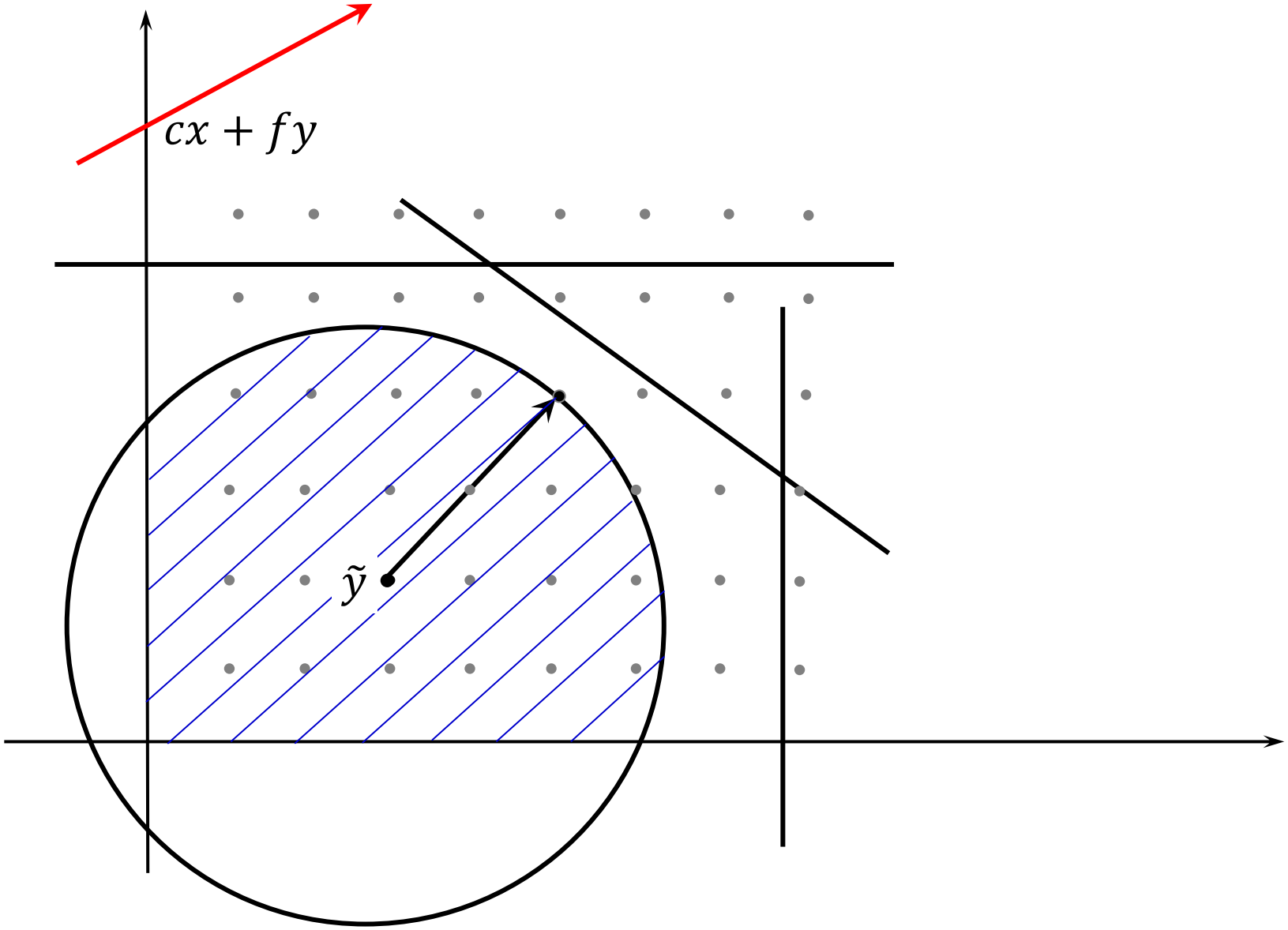
$$\sum_{j|\tilde{y}_j=0} y_j + \sum_{j|\tilde{y}_j=1} y_j(1 - y_j) \leq k;$$

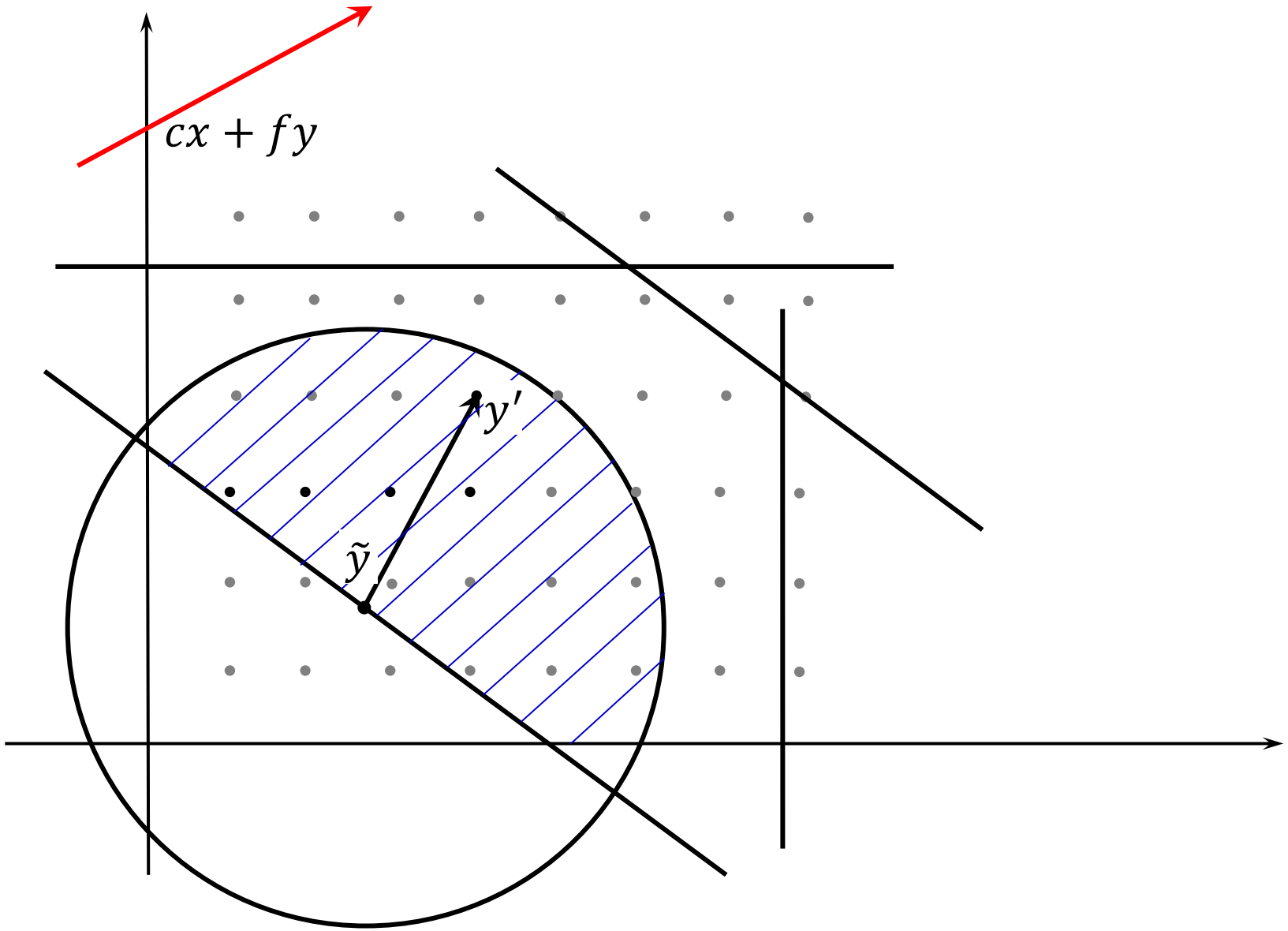
$$x \in R^n, \;\; y \in \{0,1\}^p.$$

Optimal or the best found solution in the allotted time is the result of local branching heuristic.

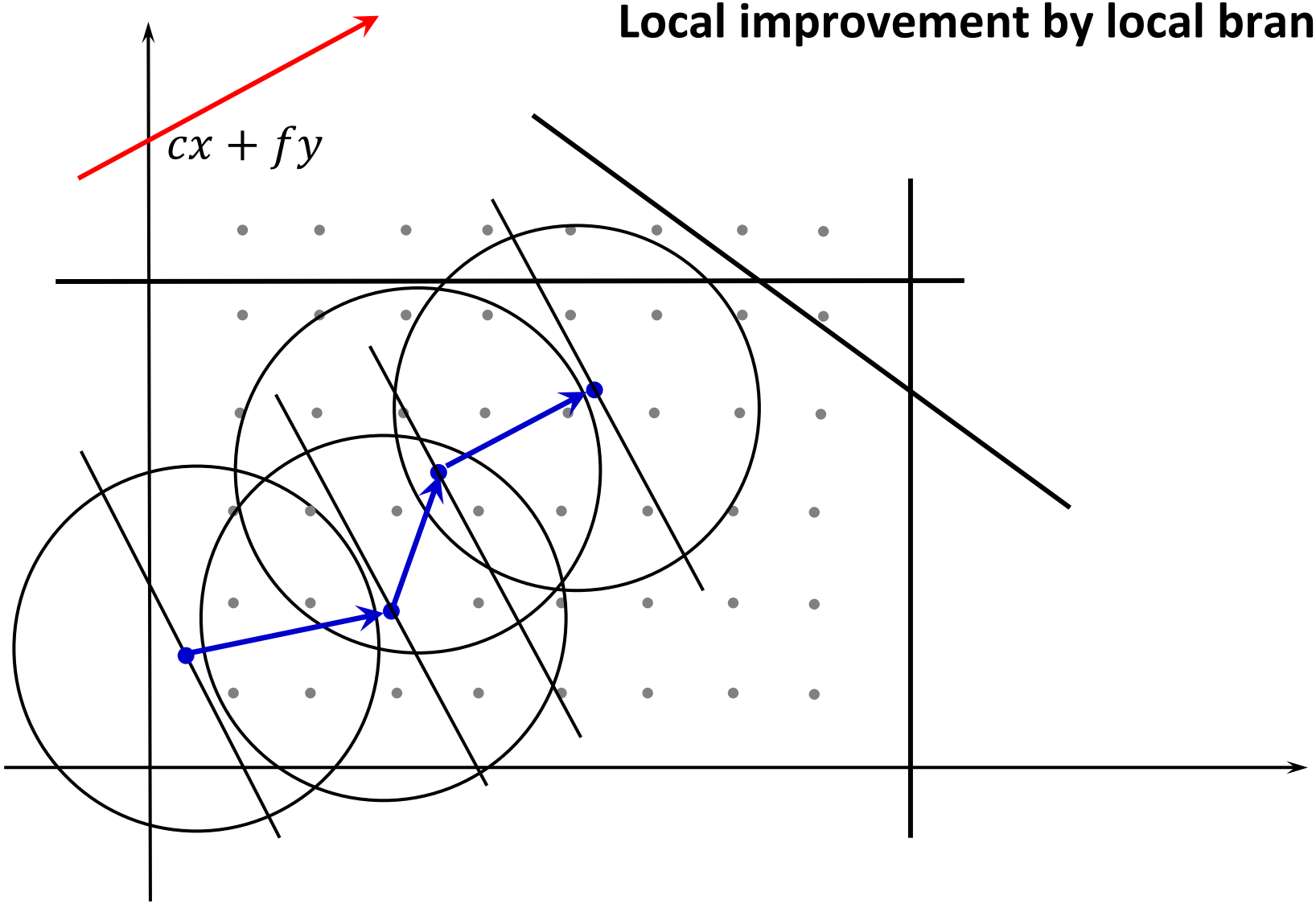
Note that $\tilde{y}$ can be feasible or not, integer or mixed integer (as result of LP relaxation). We may distinguish in importance between variables at value 0 and variables at value 1, and so on. New exact method with branching $(\ldots \leq k)\&(\ldots \geq k + 1)$.

M. Fischetti, A. Lodi. Local branching. Math.Programming. Vol. 98. P. 23–48 (2003)

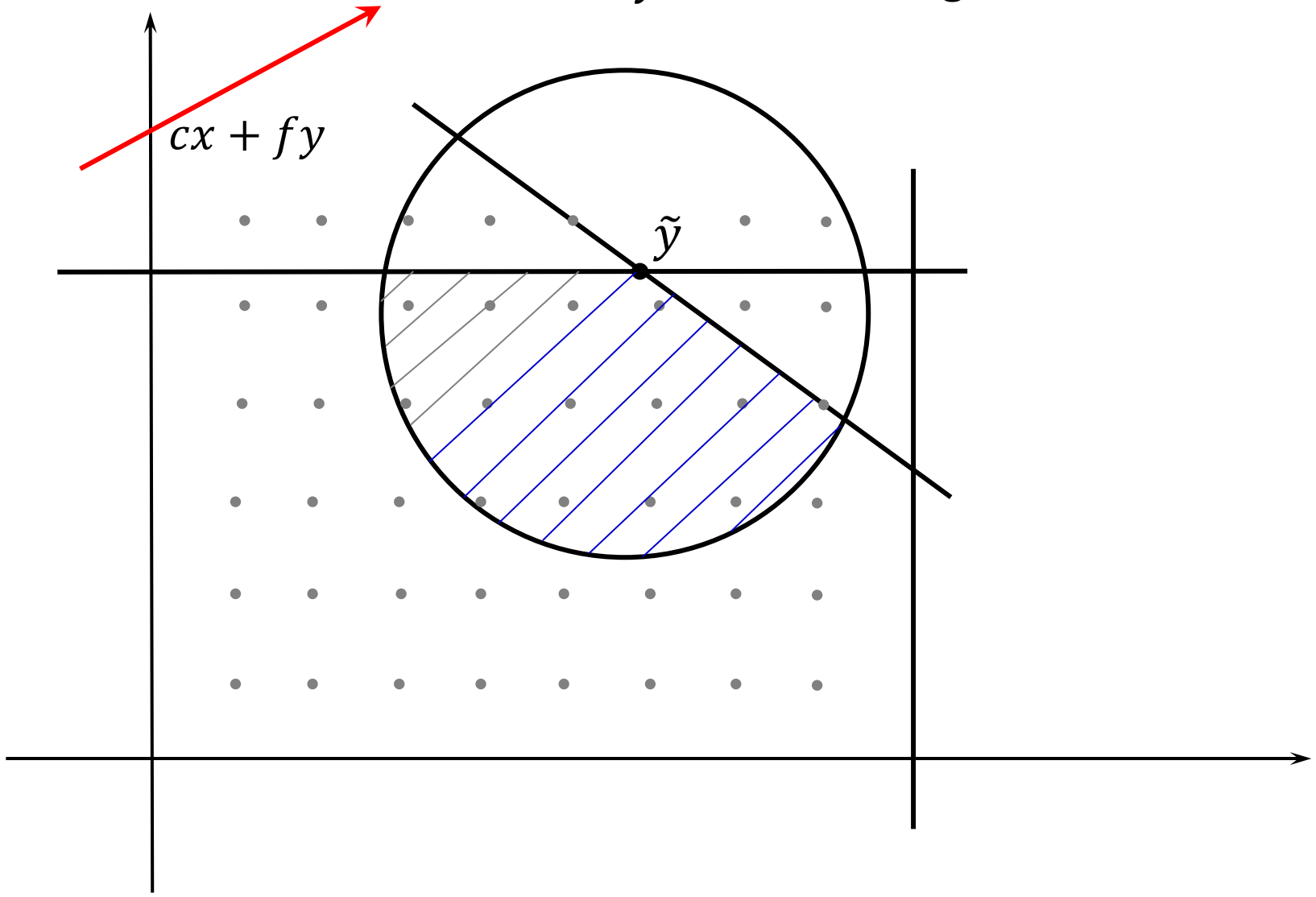

$cx + fy$
$\tilde{y}$


$cx + fy$
y'
$\tilde{y}$

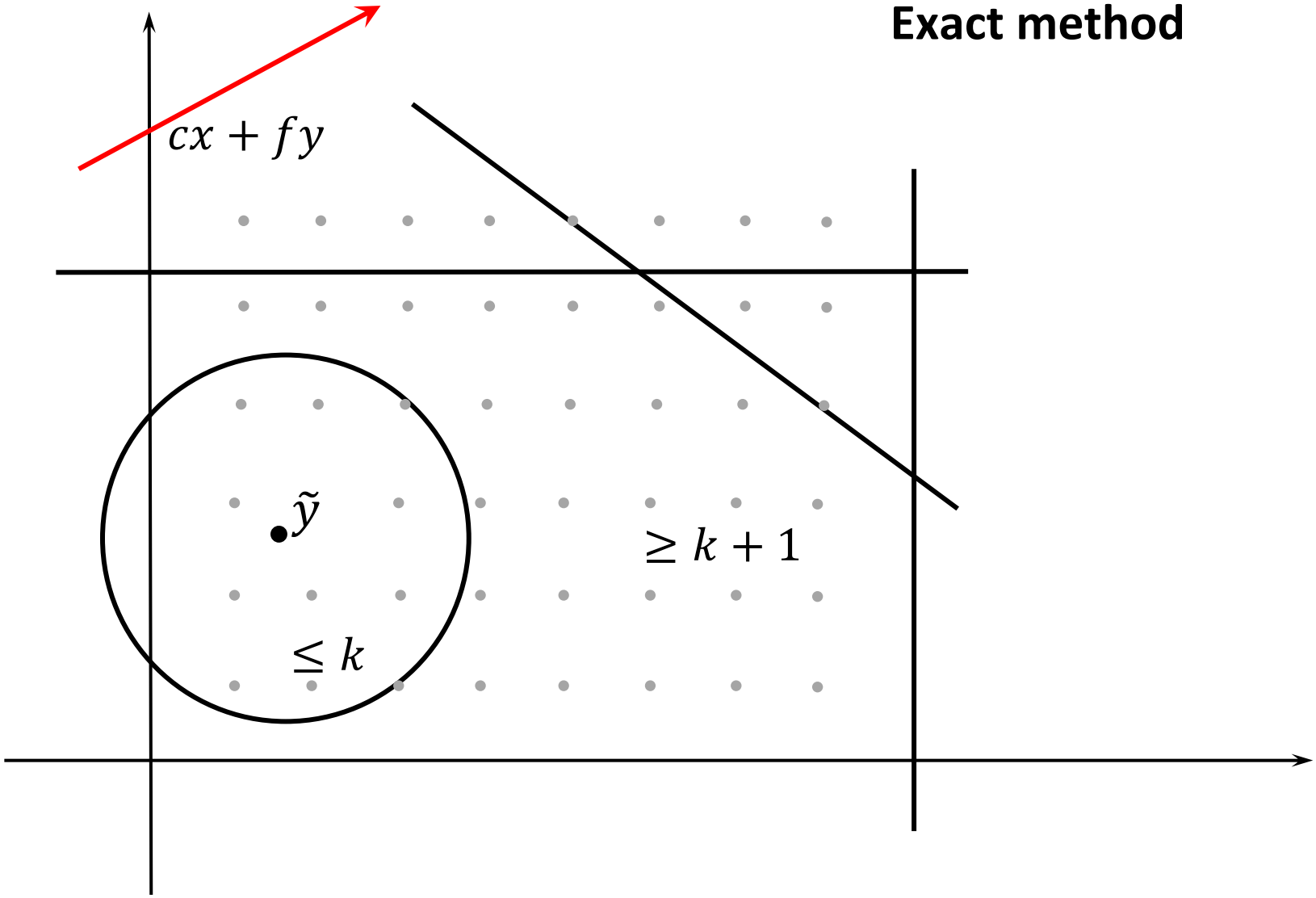
Local improvement by local branching



$\tilde{y}$ is mixed integer infeasible solution



Exact method



Large neighborhoods

The travelling salesman problem:

We deal with cities, $n = 2k$ and the set of cities is divided by two subsets:

$$x_1, x_2, \dots, x_k \quad \text{and} \quad y_1, y_2, \dots, y_k.$$

We wish to travel

$$y_1 \to x_{\tau_1} \to y_2 \to x_{\tau_2} \to \cdots k_k \to x_{\tau_k} \to y_1$$

by arbitrary permutation $\tau = (\tau_1, \tau_2, \dots, \tau_k)$.

The neighborhood $N(y_1, \dots, y_k)$ contains such tours for all permutations τ;

$$|N(y_1, \dots, y_k)| = k!$$

We need the best solution in $N(y_1, \dots, y_k)$.

We introduce the variables

$$z_{ij} = \begin{cases} 1, & \text{if } x_i \text{ between } y_j,\ y_{j+1} \\ 0, & \text{otherwise} \end{cases}$$

and the distance matrix $d_{ij} = c(y_j, x_i) + c(x_i, y_{j+1})$.

The assignment problem

$$\min \sum_{i=1}^{k} \sum_{j=1}^{k} d_{ij} z_{ij}$$

$$\sum_{i=1}^{k} z_{ij} = 1, \quad j = 1, \dots, k;$$

$$\sum_{j=1}^{k} z_{ij} = 1, \quad i = 1, \dots, k;$$

$$z_{ij} \in \{0,1\}, \qquad i, j = 1, \dots, k.$$

produces the best neighboring solution in polynomial time.

MIP Inside of Local Search

The knapsack problem with a color constraint

$$\max \sum_{i \in I} w_i x_i$$

$$\sum_{j \in J} z_j \leq b; \qquad \text{(number of colors)}$$

$$x_i \leq z_j, \quad j \in K_i, \quad i \in I$$

$$x_i, z_j \in \{0,1\}, \quad i \in I, \quad j \in J.$$

K_i is the set of colors for item i, $i \in I$.

For a feasible solution (x^0, z^0), we put $K^0 = \bigcup\{K_i \mid x_i = 1\}$ and define two neighborhoods:

k-ItemAdd neighborhood. We collect all items with at most k additional colors to K^0 and select one of them, say i'. Then we create a subset $K' = K^0 \cup K_{i'}$ and a subset of items $I' = \{i \in I \mid K_i \subseteq K'\}$. The neighboring solution is defined as the optimal solution to the problem with the following restriction: $x_i = 0$, $i \notin I'$.

k-ColorDel neighborhood. We select a subset of colors $K' \subseteq K^0, |K'| \leq k$. The neighboring solution is defined as the optimal solution to the problem with the following restriction: $z_j = 1;\ j \in K^0 \backslash K'$.

The size of the neighborhoods is exponential.

Core Concepts

The p-median problem

$$\min \sum_{i \in I} \sum_{j \in J} c_{ij} x_{ij}$$

$$\sum_{i \in I} x_{ij} = 1, \quad j \in J;$$

$$x_i \geq x_{ij}, \quad j \in J, i \in I;$$

$$\sum_{i \in I} x_i = p;$$

$$x_i, x_{ij} \in \{0,1\}, \quad i \in I, j \in J.$$

The Lagrangean relaxation provides a lover bound. Core selection approach consists of choosing a subset of “promising” variables and solving a reduced PMP over them. Core selection proved to be effective in solving capacitated facility location, PMP and others.

P. Avella, M.Boccia, S.Salerno, I.Vasilyev. An aggregation heuristic for large scale *p*-median problem. Computer&OR. Vol.39(7). P.1625–1632 (2012).

Feasibility Pump

We consider a MIP problem of the form:

$$\min cx$$
$$Ax \geq b;$$
$$l \leq x \leq u;$$
$$x_j \text{ integer } \; j \in J$$

The FP starts from the solution of the LP-relaxation $\bar{x}^0$ and generates two sequences of points: $\bar{x}^k$ and $\tilde{x}^k$.

$\bar{x}^k$ is LP-feasible, but may be integer infeasible;

$\tilde{x}^k$ is integer, but necessarily LP-feasible.

At each iteration, we solve LP problem to find $\bar{x}^{k+1}$ as the nearest point in l_1-norm to $\tilde{x}^k$

$$\bar{x}^{k+1} := \text{agrmin} \sum_{j \in J} |x_j - \tilde{x}_j^k|$$

or in equivalent form:

$$\min \sum_{j \in J | \tilde{x}_j^k = l_j} (x_j - l_j) + \sum_{j \in J | \tilde{x}_j^k = u_j} (u_j - x_j) + \sum_{j \in J | l_j < \tilde{x}_j^k < u_j} d_j$$

$$Ax \geq b;$$

$$l \leq x \leq u;$$

$$-d_j \leq x_j - \tilde{x}_j^k \leq d_j, \qquad j \in J, \qquad l_j < \tilde{x}_j^k < u_j;$$

The point $\tilde{x}^{k+1}$ is obtained as the rounding of $\bar{x}^{k+1}$. The procedure stops if at some iteration we get a feasible integer solution or if it reaches a time (iteration) limit.

M. Fischetti, F. Glover, A. Lodi. The feasibility pump // Mathematical Programming, Vol. 104. P. 91-104 (2005)

Conclusion

Metaheuristics and exact MP methods are complementary optimization strategies in terms of the quality of solutions and the search time used to find them. Combining metaheuristics with mathematical programming tools open new lines for designing effective algorithms for real world applications.

We have discussed some important ways such as Local Branching, Core Concepts, Large Neighborhoods, but ignore combining with Machine Learning and Data Mining, Constraint Programming, Multi-Objective Optimization and other promising ideas.