Facility location problems
Discrete models and local search methods

Yuri Kochetov

Sobolev Institute of Mathematics
Novosibirsk Russia
e-mail: jkochet@math.nsc.ru

Lecture 4

Average case behavior and approximability

Content

- Empirical result (for the standard local search algorithm)
- Average case analysis for random functions on hyper cube.
- Approximate local search
- The class Guaranteed Local Optima (GLO)
- $\text{APX} = \overline{\text{GLO}}$

Emperical results

- The Standard local search algorithm is polynomial in average
- There are instances with exponential number of local optima
- For many problems local search optima are relatively close to each other

Pivoting rules

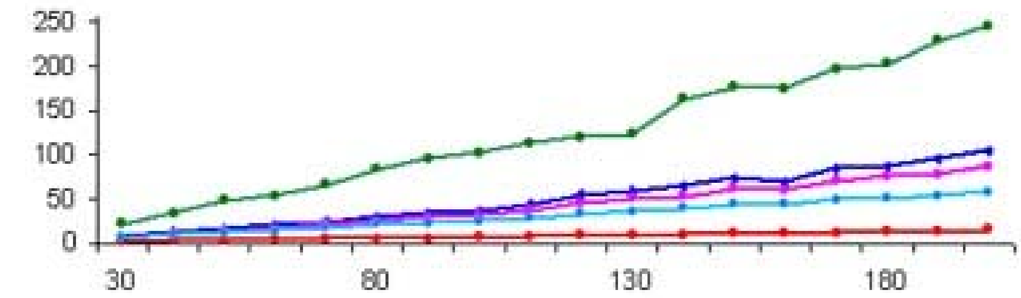
Let $N^*(s)$ be a set of neighboring solutions with better value of the objective function

$$N^*(s) = \{s' \in N(s) \mid F(s') < F(s)\}, \quad s \in Sol(x).$$

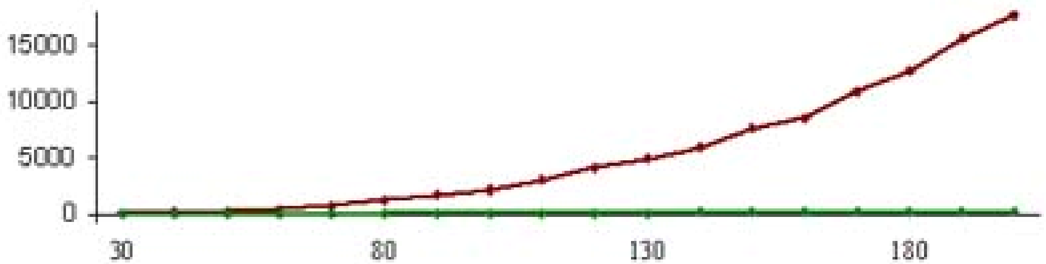
- The Best improvement rule selects the best element in the set $N^*(s)$.
- The Worst improvement rule selects the worst element in $N^*(s)$.
- The Random improvement rule picks an element in $N^*(s)$ at random.
- The First improvement rule uses the first found element in $N^*(s)$.
- The Circular rule is a variant of the First improvement rule but starts the search from the position where the previous step terminates.
- The Freedom rule select an element $s' \in N(s)$ with maximal cardinality of the set $N^*(s')$.

Computational results for the *p*-median problem with *Swap* neighborhood

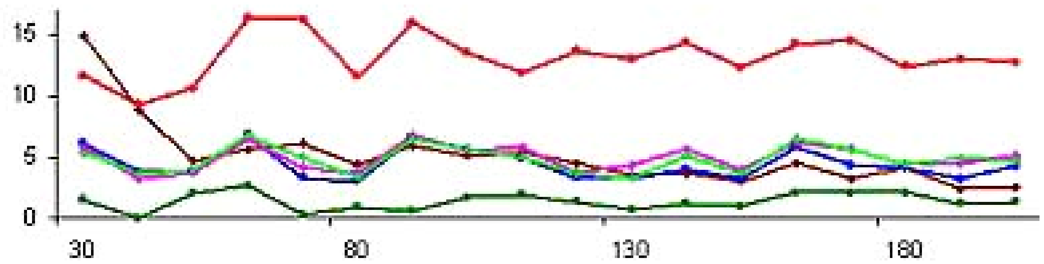
The average number of steps without *Worst* Rule, $p = n/10$



The average number of steps for the *Worst* and the *Maximal Freedom* Rule, $p = n/10$

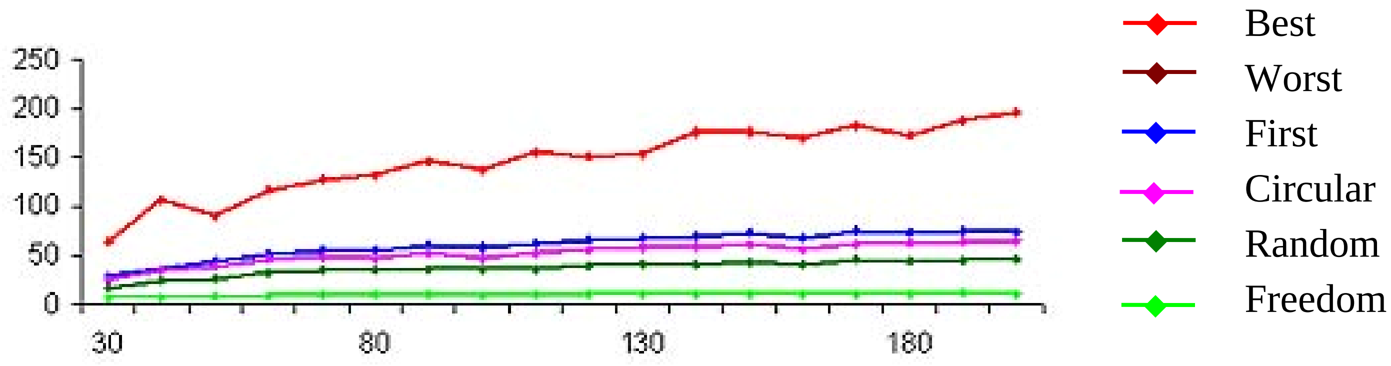


Average relative error, $p = n/10$

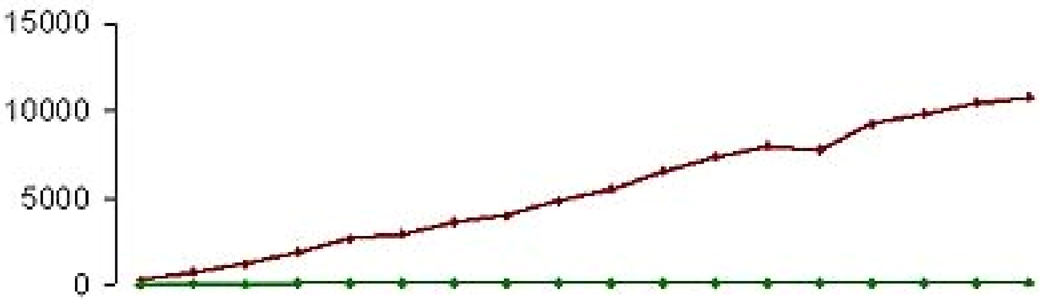


Computational results for the *p*-median problem with *Swap* neighborhood

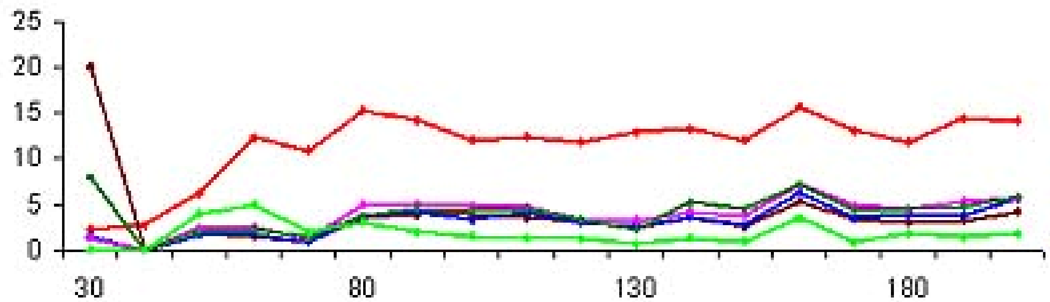
The average number of steps without *Worst* Rule, $p = 15$



The average number of steps for the *Worst* and the *Maximal Freedom* Rule, $p = 15$



Average relative error, $p = 15$



Average case analysis

Let us consider the problem min $\{F(s), s \in E^n\}$ and assume that the values of F are distinct.

Neighborhood $N(s) = \{s' \in E^n \mid d(s', s) = 1\}$.

Question: How many steps is required for the standard local search algorithm to reach local search?

For any objective function F(s) we can construct an «ordering», a list of the vertices from the best to worst function value. The random distribution we consider is that all orderings are equally likely to occur.

Theorem 4.1. [Tovey] Under the assumption that all ordering are equally likely, the expected number of steps of any local improvement algorithm is less than 1,5 en, where e denotes the logarithmic constant.

Theorem 4.2. [Tovey] Suppose the ratio of probabilities of occurrence satisfies

$$\frac{\text{Prob}[v]}{\text{Prob}[v']} \le 2^{\alpha n}$$

for all orderings v, v'. Then the expected number of steps of any local improvement algorithm is less then $(\alpha + 2)en$.

Theorem 4.3. [Tovey] Suppose the vertices of the hypercube E^n are assigned neighbors is such a way that every vertex has at most $q(n)$ neighbors, where $q(n) \ge n$ is a polynomial. Then for any probability distribution satisfying $\frac{\text{Prob}[v]}{\text{Prob}[v']} \le 2^{\alpha n}$ for all orderings v, v', the expected number of steps of any local improvement algorithm is less them $e(\alpha + 2)q(n)$.

Aproximate Local Search

Definition 4.1. We say that a feasible solution $s^{\varepsilon} \in Sol(x)$ to instance $x \in I$ is *an ε–local minimum* if $F(s^{\varepsilon}) \le (1 + \varepsilon) F(s)$ for all $s \in N(s^{\varepsilon})$.

Definition 4.2. A family of algorithms $(A_{\varepsilon})_{\varepsilon > 0}$ is *an ε–local optimization scheme* if A_{ε} produce an ε–local minimum. If the running time of algorithm A_{ε} is polynomial in the input size and $1/\varepsilon$, it is called *a fully polynomial time ε–local optimization scheme*.

Local FPTAS

Let us consider a combinatorial optimization problem with $Sol(x)$ is a family of subset of a finite ground set $E = \{1,\ldots, n\}$ and linear objective function $F(s) = \sum_{e\in S} f_e$. The goal is to find a local minimum with respect to the neighborhood N: $Sol(x)\to 2^{Sol(x)}$. We assume that this local search problem is in the class PLS and $F(s) > 0$ for all $s\in Sol(x)$.

New goal is to find an ε–local minimum for given $\varepsilon > 0$.

Algorithm ε–Local Search

1. Find $s^0 \in Sol(x)$ and put $i := 0$.
2. Put $K := F(s^i)$, $q := \frac{K\varepsilon}{2n(1+\varepsilon)}$, $f'_e := \lceil \frac{f_e}{q} \rceil q$, $e \in E$.
3. Put $j := 0$ and $s^{ij} := s^i$.
4. Repeat until $F(s^{ij}) \le K/2$

   If s^{ij} is local optimum then $s^\varepsilon := s^{ij}$, STOP

   else select better solution $s^{ij+1} \in N(s^{ij})$, $F(s^{ij+1}) < F(s^{ij})$ and put $j := j + 1$.
5. Put $s^{i+1} := s^{ij}$, $i := i + 1$ and goto 2.

Theorem 4.4. [Orlin, Punnen, Schulz] Algorithm ε–Local Search produces an ε–local optimum and its running time is polynomial in the input size and $1/\varepsilon$.

Proof. Let s^{ε} be the solution produced by the algorithm and $s \in N(s^{\varepsilon})$. Note that

$$F(s^{\varepsilon}) = \sum_{e \in s^{\varepsilon}} f_e \le \sum_{e \in s^{\varepsilon}} \lceil \frac{f_e}{q} \rceil q \le \sum_{e \in s} \lceil \frac{f_e}{q} \rceil q \le \sum_{e \in s} q(\frac{f_e}{q} + 1) \le \sum_{e \in s} f_e + nq = F(s) + nq.$$

If $F(s^{\varepsilon}) \ge K/2$ then $\dfrac{F(s^{\varepsilon}) - F(s)}{F(s)} \le \dfrac{nq}{F(s)} \le \dfrac{nq}{F(s^{\varepsilon}) - nq} \le \dfrac{2nq}{K - 2nq} = \varepsilon.$

Let us analyze the running time. Step1 is polynomial because the local search problem is in the class PLS. In each improvement move in Step 4 we get improvement at least q units. Thus the number of local search iterations at the Step 4 is $O(n(1+\varepsilon)/\varepsilon) = O(n/\varepsilon)$.

Step 2 is executed at most log $F(s^0)$ times. Thus the total number of the local search iterations is $O(n \log F(s^0)/\varepsilon)$. ■

Remark [Orlin, Punnen, Schulz] The total number of the local search iterations is $O(n^2 \varepsilon^{-1} \log n)$.

Negative Results

Theorem 4.5. If there is an algorithm that for every instance x of PLS–complete local search problem (P,N) finds in polynomial time a feasible solution s^{ε} such that

$$F(s^{\varepsilon}) \leq F(s) + \varepsilon \text{ for all } s \in (\, s^{\varepsilon})$$

for some fixed $\varepsilon > 0$, then all problems in the class PLS are polynomially solvable.

Proof. Without lost of generality we may assume that objective function is an integer–value function. For each instance x we create a new instance x' with the same set of feasible solution $Sol(x') = Sol(x)$ and new objective function $F'(s) = \sum_{e \in S} f'_e,\ s \in Sol(x')$ where $f'_e = f_e(1+\varepsilon), e \in E$.

Apply the given algorithm to the new instance x' and let s' be the resulting solution. Then, $F'(s') - F'(s) \leq \varepsilon$ for all $s \in N(s')$. Thus, $F(s') - F(s) \leq \varepsilon/(\varepsilon+1) < 1$ for all $s \in N(s')$ and s' is a local optimum for x. ■

Theorem 4.6. If a PLS–complete local search problem (P,N) has a fully polynomial time ε–local optimization scheme $(A_\varepsilon)_{\varepsilon > 0}$ such that the actual running time of A_ε is polynomial in the input size and $\log 1/\varepsilon$, then all problems in the class PLS are polynomially solvable.

Proof. We assume without lost of generality that $F(s)$ is an integer–value function. Choose $\varepsilon = 1/(nf_{\max} + 1)$, $f_{\max} = \max_{e \in E} f_e$, and apply A_ε. Note that its running time is polynomial in the input size. If s^ε is the solution returned by the algorithm, then

$$F(s^\varepsilon) \le (1 + \varepsilon)F(s) < F(s) + 1 \text{ for all } s \in N(s^\varepsilon)$$

Hence, s^ε is a local optimum. ■

Remark. For the facility location problems which we consider, these results hold.

We say that the neighborhood N is ***exact*** if each local optimum is global optimum.

Remember that a problem is called *strongly NP–hard* if it remains NP–hard even when the weights (costs) of its instances are polynomially bounded.

Theorem 4.7. [Yannakakis] Let a local search problem (P,N) is in the class PLS.

1. If P is strongly NP–hard, then N can not be exact unless P = NP.
2. If P is NP–hard, then N can not be exact unless NP = co–NP.

Theorem 4.8. [Yannakakis] Let a local search problem (P,N) is in the class PLS.

1. If the approximation of P within a factor r is strongly NP–hard, then N does not guarantee ratio r unless P = NP.
2. If the approximation of P within a factor r is NP–hard, then N does not guarantee ratio r unless P = co–NP.

Class GLO

We say that optimization problem P is *polynomially bounded* if there exists a polynomial r such that, given any instance $x \in I$ and given any feasible solution $s \in Sol(x)$, $F(s) \le r(|x|)$.

Definition 4.3. An optimal problem P has *guaranteed* local optima if there exists a polynomial time computable neighborhood N and a constant k $(0 < k \le 1)$ such that, for every instance $x \in I$, any local optimum s of x with respect to N has the property that $F(s) \le k\,\mathrm{OPT}(x)$ (for a minimization problem).

Definition 4.4. Let an instance $x \in I$ and feasible solution $s \in Sol(x)$ be given. We say that $s' \in Sol(x)$ is an *h–bounded neighbor of s* if $d(s,s') \le h$.

A neighborhood mapping N is said to be an h–bounden mapping if there exists constant h such that, given $x \in I$ and $s \in Sol(x)$, any $s' \in N(s)$ is an h–bounded neighbor of s.

Definition 4.5. Let P be a polynomially bounded optimization problem. We say that *P belongs to the class GLO* (Guaranteed Local Optima) if the following two conditions are satisfied:

– at least one feasible solution $s \in Sol(x)$ can be computed in polynomial time for every instance $x \in I$,

– there exists a constant h such that P has guaranteed local optima with respect to a suitable h–bounded neighborhood.

Examples:

- Metric p–median problem;
- Max– Satisfiability;
- Max–Cut.

Approximation preserving reduction

Definition 4.6. Let A and B be two optimization problems. A is said to be *PTAS–reducible* to B (in symbol $A \le_{PTAS} B$) if three function f, g, c exist such that:

- for any $x \in I_A$ and for any $\varepsilon \in (0,1)_Q$, (Q is the set of rational number) $f(x,\varepsilon) \in I_B$ is computable in time polynomial with respect to $|x|$;
- for any $x \in I_A$, for any $s \in Sol_B(f(x, \varepsilon))$, and for any $\varepsilon \in (0,1)_Q$, $g(x, s, \varepsilon) \in Sol_A(x)$ is computable in time polynomial with respect to both $|x|$ and $|s|$;
- c: $(0,1)_Q \to (0,1)_Q$ is computable and surjective;
- for any $x \in I_A$, for any $s \in Sol_B(f(x, \varepsilon))$, and for any $\varepsilon \in (0,1)_Q$

$$E_B(f(x, \varepsilon), s) \le c(\varepsilon) \text{ implies } E_A(x, g(x, \varepsilon)) \le \varepsilon,$$

where $E(x,s)$ is the relative error of s for x, $E(x,s) = \dfrac{|F(s) - \mathrm{OPT}(x)|}{\max\{F(s), \mathrm{OPT}(x)\}}$.

Closure of GLO under approximation preserving reductions

Given two optimization problems A and B. If $A \le_{PTAS} B$ and $B \in$ APX ($B \in$ PTAS) then $A \in$ APX ($A \in$ PTAS).

If $\mathcal{C}$ is a class of optimization problems, then by $\overline{\mathcal{C}}$ we denote the closure of $\mathcal{C}$ under PTAS reductions, that is, the set of problems

$$\overline{\mathcal{C}} = \{ A \mid \exists\, B \in \mathcal{C} \text{ such that } A \le_{PTAS} B \}$$

Theorem 4.9. [Ausiello, Protasi] $\overline{\text{GLO}} = \text{APX}$.