Volume 191, Issue 3, 16 December 2008 ISSN 0377-2217

ELSEVIER

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

Available online at

ScienceDirect
www.sciencedirect.com

Feature Clusters:

Variable Neighbourhood Search
Guest Editors: P. Hansen, N. Mladenović and J.A. Moreno Perez

Continuous Optimization in the Industry
Guest Editors: T. Illés, M.A. Lopez, T. Terlaky, J. Vörös and G.-W. Weber

The Car Sequencing Problem
Guest Editors: C. Solnon, V.D. Cung, A. Nguyen and C. Artigues

Performance Evaluation of QoS-Aware Heterogeneous Systems
Guest Editor: C. Skianis

Closed-Loop Supply Chain
Guest Editor: S.D. Flapper

LAST ISSUE OF THIS VOLUME

www.elsevier.com/locate/ejor

Available online at www.sciencedirect.com

ScienceDirect

European Journal of Operational Research 191 (2008) 736–752

EUROPEAN JOURNAL OF OPERATIONAL RESEARCH

www.elsevier.com/locate/ejor

Complexity of local search for the *p*-median problem

Ekaterina Alekseeva, Yuri Kochetov *, Alexander Plyasunov

Sobolev Institute of Mathematics, Novosibirsk, Russia

Received 15 April 2006; accepted 15 December 2006
Available online 16 February 2007

Abstract

We study the complexity of finding local minima for the *p*-median problem. The relationship between Swap local optima, 0–1 local saddle points, and classical Karush–Kuhn–Tucker conditions is presented. It is shown that the local search problems with some neighborhoods are tight PLS-complete. Moreover, the standard local descent algorithm takes exponential number of iterations in the worst case regardless of the tie-breaking and pivoting rules used. To illustrate this property, we present a family of instances where some local minima may be hard to find. Computational results with different pivoting rules for random and Euclidean test instances are discussed. These empirical results show that the standard local descent algorithm is polynomial in average for some pivoting rules.

Keywords: Local search; PLS-completeness; Pivoting rules; Karush–Kuhn–Tucker conditions; 0–1 local saddle points; Pseudo-Boolean functions

1. Introduction

In the *p*-median problem we are given a set $I = \{1,\dots,n\}$ of potential locations for p facilities, a set $J = \{1,\dots,m\}$ of customers, and a matrix (g_{ij}), $i \in I$, $j \in J$ of transportation costs for servicing the customers by the facilities. The goal is to find a subset $S \subset I$, $|S| = p$, that minimizes the objective function $F(S) = \sum_{j\in J}\min_{i\in S} g_{ij}$. It is a well-known combinatorial problem, which is NP-hard in a strong sense. Moreover, assuming $P \neq \text{NP}$, no polynomial time algorithm can guarantee a relative error at most $2^{q(n,m)}$ for any fixed polynomial q and all instances of the *p*-median problem (Nemhauser and Wolsey, 1988). In other words, this problem does not belong to the class APX, and finding good approximation is as hard as determining an optimal solution. In what follows, iterative local search methods seem the most promising for the problem. A recent survey on the state of the art in this area can be found in Mladenović et al. (2007).

We say that a neighborhood is polynomially searchable if exists a polynomial time algorithm with the following properties. Given an instance and a solution, the algorithm determines whether solution is a local optimum, and if it is not, the algorithm outputs a neighbor with strictly better value of objective function. The

[*] Corresponding author.
E-mail addresses: ekaterina2@math.nsc.ru (E. Alekseeva), jkochet@math.nsc.ru (Y. Kochetov), apljas@math.nsc.ru (A. Plyasunov).

0377-2217/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2006.12.063

complexity class, called PLS (polynomial time local search), contains the problems whose neighborhood can be searched in polynomial time (Johnson et al., 1988). Many important local search problems are complete for the class PLS under an appropriately defined reduction. If a local optimum for such a complete problem can be found in polynomial time by whatever means, then for all problems in the class PLS a local optimum can be found in polynomial time. "*This is generally not believed to be true, as it would require a general approach for finding local optima at least as clever as the ellipsoid algorithm, since linear programming with the simplex neighborhood is in PLS*" (Vredeveld and Lenstra, 2003). On the other hand, if a PLS problem is NP-hard, then NP = co-NP (Johnson et al., 1988). So, it is very unlikely that the class PLS contains an NP-hard problem. Therefore, the local search problems may be not so difficult.

Many local search heuristics, for example, Variable Neighborhood Search, GRASP, Memetic algorithms, use standard local descent procedures and focus on the local optima only. In this paper, we study the complexity of finding a local minimum for polynomially searchable neighborhoods for the p-median problem. We show the relations between Karush–Kuhn–Tucker conditions, Swap-optimal solutions and Swap-saddle points. Moreover, we present a sufficient condition when the p-median problem with polynomially searchable neighborhood is PLS-complete. Several polynomial neighborhoods are introduced, and it is shown that in the worst case the standard local descent algorithm takes exponential number of steps with each neighborhood regardless of the tie–breaking and pivoting rules used. We consider several pivoting rules and present computational results for random and Euclidean test instances. We note that the number of steps grows as a linear function for the pivoting rules *best improvement* and *first improvement* and grows as a superlinear function for the *worst improvement* rule if $p = [\alpha n],\ 0 \leqslant \alpha \leqslant n$. A theoretical explanation of this phenomenon is discussed.

The paper is organized as follows. In Section 2, we define some neighborhoods. In Section 3, we discuss a one to one correspondence between Swap-optimal solutions and 0–1 local saddle points for Lagrange function. In Section 4, the PLS-completeness of the p-median problem with several neighborhoods is established. In Section 5, we present a family of instances where standard local descent algorithm takes exponential number of steps to reach a Swap local minimum. We define approximate local optima in Section 6 and show the complexity of corresponding local search problems. Pivoting rules are described in Section 7. The running time of the local descent algorithm is studied experimentally in Section 8. Conclusions and further research directions are discussed in Sections 9.

2. Neighborhoods

The Swap neighborhood is one of the effective and efficient neighborhoods for the p-median problem (Resender and Werneck, 2003). It contains all subsets $S' \subset I,\ |S'| = p$, with the Hamming distance from S' to S is equal 2. Similarly, the k-Swap neighborhood is the set of all feasible solutions with Hamming distance from S' to S at most k. Finding the best element in this neighborhood is time consuming for large k. So, this neighborhood is interesting for theoretical study only.

The Kernighan–Lin neighborhood (KL) is a subset of the k-Swap neighborhood. It consists of k elements, $k = \min\{p, n-p\}$, and can be described by the following steps (Kernighan and Lin, 1970).

Step 1. Choose two elements $i_{\text{ins}} \in I \setminus S$ and $i_{\text{rem}} \in S$ such that $F(S \cup \{i_{\text{ins}}\} \setminus \{i_{\text{rem}}\})$ is minimal even if it is greater than $F(S)$.

Step 2. Perform swap of i_{rem} and i_{ins}.

Step 3. Repeat Steps 1 and 2 k times such that the elements cannot be chosen to be inserted in S or removed from S if they have been used at one of the previous iterations of Steps 1 and 2.

The sequence $\{(i^{\tau}_{\text{ins}}, i^{\tau}_{\text{rem}})\}_{\tau \leqslant k}$ defines k neighbors S_τ for solution S. We say that S is a local minimum with respect to KL-neighborhood if $F(S) \leqslant F(S_\tau)$ for all $\tau \leqslant k$. The neighborhood $\text{KL}_1(S)$ is defined to be a subset of KL(S) which contains the first element only, $S_\tau, \tau = 1$. By definition, $\text{KL}_1(S) \subset \text{Swap}(S)$.

The Fiduccia–Mattheyses neighborhood (FM) is defined as the KL-neighborhood with a different rule for the choice of elements i_{ins} and i_{rem} at the Step 1 (Fiduccia and Mattheyses, 1982). This step consists of two stages. At first, we select $i_{\text{rem}} \in S$ such that $F(S \setminus \{i_{\text{rem}}\})$ is minimal. At the second stage, we find $i_{\text{ins}} \in I \setminus S$

such that $F(S \cup \{i_{\text{ins}}\} \setminus \{i_{\text{rem}}\})$ is minimal. It defines the sequence $S_\tau, \tau \leqslant k$, of neighbors for the solution S. The neighborhood $\text{FM}_1(S)$ contains only the first element from this sequence.

We say that neighborhood N_1 is stronger than neighborhood N_2 ($N_2 \preceq N_1$) if every N_1-optimum is N_2-optimum. It is easy to verify that

$$\text{FM}_1 \preceq \text{Swap} \preceq \text{KL}_1 \preceq \text{KL},$$
$$\text{KL}_1 \preceq \text{Swap} \preceq k\text{-Swap},$$
$$\text{FM}_1 \preceq \text{FM}.$$

For any constant $k > 0$, all neighborhoods are polynomial. The neighborhoods Swap and KL_1 are equivalent with respect to the relation $\preceq$ and neighborhood FM_1 is the most weak.

3. Local saddle points

In this section we show that there is a strong connection between Swap-optima and the local saddle points for the Lagrange function. Let us rewrite the p-median problem as the minimization problem for a pseudo-Boolean function on $(n-p)$-layer of the hypercube. For a given vector $g_i,\ i \in I$ with ranking

$$g_{i_1} \leqslant g_{i_2} \leqslant \cdots \leqslant g_{i_m},$$

let us introduce a vector $\Delta g_i, i = 0, \ldots, m-1$ in the following way:

$$\Delta g_0 = g_{i_1}, \quad \Delta g_l = g_{i_{l+1}} - g_{i_l}, \quad 1 \leqslant l < m.$$

For an arbitrary vector $y_i \in \{0,1\},\ i \in I,\ y \neq (1, \ldots, 1)$, the following statement holds (Beresnev et al., 1978, Lemma 1.1):

$$\min_{i \mid y_i = 0} g_i = \sum_{l=0}^{m-1} \Delta g_l y_{i_1} \cdots y_{i_l}.$$

Similar, we introduce the ranking $i_i^j, \ldots, i_m^j$ which is generated by the column j of the matrix (g_{ij}), $i \in I$, $j \in J$:

$$g_{i_1^j} \leqslant g_{i_2^j} \leqslant \cdots \leqslant g_{i_m^j}, \quad j \in J.$$

Now one can get a pseudo-Boolean function for the p-median problem:

$$P(y) = \sum_{j \in J} \sum_{l=0}^{m-1} \Delta g_{lj} y_{i_1^j} \cdots y_{i_l^j}.$$

An optimal solution $y_i^*,\ i \in I$ for the minimization problem for this pseudo-Boolean function on the $(n-p)$-layer of the hypercube gives us an optimal solution S^* for the p-median problem. More exactly, $y_i^* = 0$ if and only if $i \in S^*$ (Beresnev et al., 1978, Theorem 3.2). Note that $P(y)$ has positive terms only. So, we can rewrite the p-median problem as the minimization problem for a pseudo-Boolean function on $(n-p)$-layer of the hypercube:

$$\text{Minimize} \quad P(y) = \sum_{j \in J'} a_j \prod_{i \in I_j} y_i$$
$$\text{s.t.} \quad \sum_{i \in I} y_i = n - p, \quad y_i \in \{0,1\}, \quad i \in I,$$

where $a_j \geqslant 0,\ I_j \subset I,\ j \in J' = \{1, \ldots, n \times m\}$. There is a one-to-one correspondence between feasible solutions of the p-median problem and feasible solutions of this problem. In fact, $i \in S$ iff $y_i = 0$ for all $i \in I$. Moreover, $F(S) = P(y)$. So, we can reconstruct y from S and get $y(S)$ and, vice versa, get $S(y)$ by y. Therefore, S is Swap-optimum iff $y(S)$ is Swap-optimum.

Let us replace the Boolean constraints $y_i \in \{0,1\}$ by $0 \leqslant y_i \leqslant 1$. The Lagrange function with multipliers $\lambda, \mu_i \geqslant 0, \sigma_i \geqslant 0, i \in I$ is as follows:

$$L(y,\lambda,\mu,\sigma) = P(y) + \lambda\left(n - p - \sum_{i\in I} y_i\right) + \sum_{i\in I}\sigma_i(y_i - 1) - \sum_{i\in I}\mu_i y_i.$$

Let $P'_i(y)$ denote the first derivative of $P(y)$ with respect to the variable y_i. The correspondent Karush–Kuhn–Tucker conditions (KKT) are

$$\frac{\partial L}{\partial y_i}(y,\lambda,\mu,\sigma) = P'_i(y) - \lambda + \sigma_i - \mu_i = 0, \quad i = 1,\dots,n,$$
$$\sum_{i\in I} y_i = n - p, \quad 0 \leqslant y_i \leqslant 1, \quad i \in I,$$
$$\sigma_i(y_i - 1) = 0, \quad \mu_i y_i = 0, \quad i \in I.$$

The vector $(y^*,\lambda^*,\mu^*,\sigma^*)$ is called the saddle point with respect to Swap neighborhood or Swap-saddle point if

$$L(y^*,\lambda,\mu,\sigma) \leqslant L(y^*,\lambda^*,\mu^*,\sigma^*) \leqslant L(y,\lambda^*,\mu^*,\sigma^*) \qquad (1)$$

for all $\lambda, \mu \geqslant 0, \sigma \geqslant 0$ and all Boolean vectors $y \in \mathrm{Swap}(y^*)$.

Theorem 1. *For any feasible solution S^* of the p-median problem the following properties are equivalent:*

(i) *There are the multipliers $\lambda^*, \mu_i^* \geqslant 0, \sigma_i^* \geqslant 0, i \in I$ such that the vector $(y(S^*),\lambda^*,\mu^*,\sigma^*)$ is the Swap-saddle point of the function L.*
(ii) *S^* is Swap-optimum.*
(iii) *$y(S^*)$ satisfies the KKT conditions.*

Proof

1. Let us check (i) ⇒ (ii). Let $(y(S^*),\lambda^*,\mu^*,\sigma^*)$ be a Swap-saddle point. Put $y^* = y(S^*)$. Using the left part of (1) we get

$$L(y^*,\lambda^*,\mu^*,\sigma^*) = \sup_{\lambda,\mu\geqslant 0,\sigma\geqslant 0} L(y^*,\lambda,\mu,\sigma) = P(y^*). \qquad (2)$$

Trivially, the left part of (2) holds. Now we show the right part of (2). If $y_i^* - 1 < 0$ or $y_i^* > 0$, then the corresponding Lagrange multiplier σ_i^* or μ_i^* vanishes, otherwise the left part of (2) is not satisfied. Therefore, the complementary slackness conditions,

$$\lambda^*\left(\sum_{i\in I} y_i^* - n + p\right) = 0, \quad \sigma_i^*(y_i^* - 1) = 0, \quad \mu_i^* y_i^* = 0, \quad i \in I,$$

hold, and we obtain (2). So, we have

$$P(y^*) \leqslant L(y,\lambda^*,\mu^*,\sigma^*) \quad \text{for all } y \in \mathrm{Swap}(y^*).$$

Since each $y \in \mathrm{Swap}(y^*)$ is a feasible solution of the problem, we have

$$F(S^*) = P(y^*) \leqslant P(y) + \lambda^*\left(\sum_{i\in I} y_i - n + p\right) + \sum_{i\in I}\sigma_i^*(y_i - 1) - \sum_{i\in I}\mu_i^* y_i \leqslant P(y) = F(S(y)),$$

where $S(y) \in \mathrm{Swap}(S^*)$. Therefore, S^* is Swap-optimum.

2. We now show (ii) ⇒ (iii). Let us consider a Swap-optimum S^*. Boolean vector $y^* = y(S^*)$ satisfies $\sum_{i\in I} y_i^* = n - p$ and is a Swap-optimum of $P(y)$. We need to find multipliers $\lambda^*, \mu_i^* \geqslant 0$, $\sigma_i^* \geqslant 0$, $i \in I$, such that the vector $(y^*,\lambda^*,\mu^*,\sigma^*)$ satisfies the KKT conditions. Let $P''_{i_0 i_1}(y)$ be the second derivative of $P(y)$ with respect to y_{i_0} and y_{i_1}. Put

$$\Delta^{i_0}_{-i_1}(y) = P'_{i_0}(y)y_{i_0} - P''_{i_0i_1}(y)y_{i_0}y_{i_1},$$
$$\Delta^{i_1}_{-i_0}(y) = P'_{i_1}(y)y_{i_1} - P''_{i_0i_1}(y)y_{i_0}y_{i_1},$$
$$\Delta_{-i_0-i_1}(y) = P(y) - P''_{i_0i_1}(y)y_{i_0}y_{i_1} - \Delta^{i_0}_{-i_1}(y) - \Delta^{i_1}_{-i_0}(y).$$

Hence,

$$P(y) = P''_{i_0i_1}(y)y_{i_0}y_{i_1} + \Delta^{i_0}_{-i_1}(y) + \Delta^{i_1}_{-i_0}(y) + \Delta_{-i_0-i_1}(y). \tag{3}$$

Suppose $y \in \mathrm{Swap}(y^*)$, $y^*_{i_0} = 0$, $y^*_{i_1} = 1$, $y_{i_0} = 1$, $y_{i_1} = 0$, and $y_i = y^*_i$ for all $i \neq i_0, i_1$. Combining this with (3), we get

$$P(y^*) = \Delta^{i_1}_{-i_0}(y^*) + \Delta_{-i_0-i_1}(y^*) = P'_{i_1}(y^*) + \Delta_{-i_0-i_1}(y^*),$$
$$P(y) = \Delta^{i_0}_{-i_1}(y) + \Delta_{-i_0-i_1}(y) = P'_{i_0}(y^*) + \Delta_{-i_0-i_1}(y^*).$$

So,

$$P(y^*) - P(y) = P'_{i_1}(y^*) - P'_{i_0}(y^*). \tag{4}$$

Since y^* is Swap-optimum, we have

$$P'_{i_1}(y^*) - P'_{i_0}(y^*) \leqslant 0. \tag{5}$$

Consider indices i^*_0, i^*_1 such that

$$P'_{i^*_0}(y^*) = \min_{i:y^*_i=0} P'_i(y^*), \qquad P'_{i^*_1}(y^*) = \max_{i:y^*_i=1} P'_i(y^*).$$

Substituting i^*_0, i^*_1 in (5), we get $P'_{i^*_1}(y^*) \leqslant P'_{i^*_0}(y^*)$. Put $\lambda^* \in [P'_{i^*_1}(y^*), P'_{i^*_0}(y^*)]$ and

$$\mu^*_i = \begin{cases} P'_i(y^*) - \lambda^* \geqslant 0 & \text{if } y^*_i = 0,\\ 0 & \text{otherwise,}\end{cases}$$
$$\sigma^*_i = \begin{cases} \lambda^* - P'_i(y^*) \geqslant 0 & \text{if } y^*_i = 1,\\ 0 & \text{otherwise.}\end{cases}$$

We have $\mu^* \geqslant 0, \sigma^* \geqslant 0$ so that the complementary slackness conditions

$$\lambda^*\left(\sum_{i\in I} y^*_i - n + p\right) = 0, \quad \sigma^*_i(y^*_i - 1) = 0, \quad \mu^*_i y^*_i = 0, \quad i \in I$$

are satisfied. Also,

$$\frac{\partial L}{\partial y_i}(y^*, \lambda^*, \mu^*, \sigma^*) = P'_i(y^*) - \lambda^* + \sigma^*_i - \mu^*_i = 0, \quad i = 1, \ldots, n.$$

This proves (ii) ⇒ (iii).

3. Finally, we show (iii) ⇒ (i). Since the complementary slackness conditions hold, it follows that

$$L(y^*, \lambda^*, \mu^*, \sigma^*) = P(y^*).$$

For $y \in \mathrm{Swap}(y^*)$, $y^*_{i_0} = 0$, $y^*_{i_1} = 1$, $y_{i_0} = 1$, $y_{i_1} = 0$, and $y_i = y^*_i$ for all $i \neq i_0, i_1$, we have

$$L(y, \lambda^*, \mu^*, \sigma^*) = P(y) + \lambda^*\left(\sum_{i\in I} y_i - n + p\right) + \sum_{i\in I}\sigma^*_i(y_i - 1) - \sum_{i\in I}\mu^*_i y_i = P(y) - \sigma^*_{i_1} - \mu^*_{i_0}.$$

Since $P'_i(y) - \lambda + \sigma_i - \mu_i = 0$, $i = 1, \ldots, n$, and the complementary slackness conditions hold, we have $\sigma^*_{i_1} = \lambda^* - P'_{i_1}(y^*)$, $\mu^*_{i_0} = P'_{i_0}(y^*) - \lambda^*$. It follows that:

$$L(y, \lambda^*, \mu^*, \sigma^*) = P(y) + P'_{i_1}(y^*) - P'_{i_0}(y^*).$$

Combining this with (4), we get

$$L(y^*,\lambda^*,\mu^*,\sigma^*) = P(y^*) = P(y) + P'_{i_1}(y^*) - P'_{i_0}(y^*) = L(y,\lambda^*,\mu^*,\sigma^*).$$

So, we have got the right part of (1). Note that

$$L(y^*,\lambda,\mu,\sigma) = P(y^*) + \lambda\left(\sum_{i\in I} y_i^* - n + p\right) + \sum_{i\in I}\sigma_i(y_i^* - 1) - \sum_{i\in I}\mu_i y_i^* = P(y^*) - \sum_{i:y_i^*=0}\sigma_i - \sum_{i:y_i^*=1}\mu_i \leqslant P(y^*)$$
$$= L(y^*,\lambda^*,\mu^*,\sigma^*),$$

which completes the proof. □

4. Local search problems

Let us recall the formal definition of optimization problem (Ausiello et al., 1999). *An optimization problem* OP is defined by the quadruple $<\mathscr{I}, \mathrm{Sol}, F,$ goal $>$, where

(1) $\mathscr{I}$ is the set of instances of OP;
(2) Sol is a function that associates to any input instance $x \in \mathscr{I}$ the set of feasible solutions of $\mathrm{Sol}(x)$;
(3) F is the objective function that, for every pair (s,x), where $s \in \mathrm{Sol}(x)$, assigns an integer $F(s,x)$;
(4) *goal* $\in \{\min, \max\}$ specifies whether OP is a maximization or a minimization problem. The problem is: given an instance x, to find an optimal solution $s \in \mathrm{Sol}(x)$.

Definition 1. *A local search problem Π* is the pair (OP, N), where OP is the optimization problem and N is the neighborhood, i.e., N is a function that assigns a set $N(s,x) \subseteq \mathrm{Sol}(x)$ of neighboring solutions for every pair (x,s). The set $N(s,x)$ is called the neighborhood of the feasible solution s. The local search problem is: given an instance x, compute a locally optimal solution s^*, i.e., a solution that has no better neighbor.

We will assume that for each instance x its solutions $s \in \mathrm{Sol}(x)$ have length bounded by a polynomial in the length of x.

In Definition 1, we allow the use of any algorithm whatsoever, not necessarily a local search algorithm. It is important to make a distinction between the complexity of the local search problem itself on the one hand and the complexity of the local search heuristic on the other hand (Johnson et al., 1988). In other words, we make a distinction between the complexity of finding local optima by any means and the complexity of finding local optima by the standard local search algorithm. Therefore, if the local search heuristic takes an exponential number of iteration, it does not preclude from finding local optima by other methods in polynomial time.

A nontrivial example of the local search problem is the linear programming problem. It can be viewed as a local search problem. The solutions are the vertices of a polytope and the neighborhood is given by edges of the polytope. The standard local search algorithm is the classical simplex method. It takes an exponential number of iteration in the worst case for the most pivoting rules. However, optimal solution can be found in polynomial time by other methods (Papadimitriou and Steiglitz, 1982).

Definition 2 (Yannakakis, 1997). A local search problem Π is in the class PLS if there are three polynomial-time algorithms A, B, C with following properties:

(1) Given a string x, algorithm A determines whether x is an instance ($x \in \mathscr{I}$), and in this case it produces some solution $s_0 \in \mathrm{Sol}(x)$;
(2) Given an instance $x \in \mathscr{I}$ and a string s, algorithm B determines whether $s \in \mathrm{Sol}(x)$ and if so, B computes the cost $F(s,x)$ of the solution s;
(3) Given an instance $x \in \mathscr{I}$ and a solution s, algorithm C determines whether s is a local optimum, and if it is not, C outputs a neighbor $s' \in N(s,x)$ with better cost.

This definition leads directly to a local descent algorithm, which starts from the initial solution s_0 generated by algorithm A, and then applies repeatedly algorithm C until it reaches a local optimum. The precise algorithm is determined by the pivoting rule chosen. For a current solution that is not a local optimum, the pivoting rule selects neighboring solution with better value of the objective function. Let us introduce the new complexity class P_{PLS} as class P in the theory of NP-completeness.

Definition 3. A local search problem $\Pi = (\text{OP}, N)$ belongs to the class P_{PLS} if it is in PLS and there exists a polynomial time algorithm that for any instance $x \in \mathscr{I}$ returns an N-optimal solution $s \in \text{Sol}(x)$.

The class P_{PLS} is the natural efficiently solvable part of the class PLS. The relationship between the classes P_{PLS} and PLS is fundamental for the theory of local search problems. Obviously, the global optimum is the local one for an arbitrary neighborhood. Hence, if $P_{\text{PLS}} \neq \text{PLS}$ then $P \neq \text{NP}$.

Definition 4 (Yannakakis, 1997). Let Π_1 and Π_2 be two local search problems. A PLS-reduction from Π_1 to Π_2 consists of two polynomial time computable functions h and g such that:

(1) h maps instances x of Π_1 to instances $h(x)$ of Π_2.
(2) g maps pairs (solution of $h(x), x$) to solutions of x.
(3) For all instances x of Π_1, if s a local optimum for instance $h(x)$ of Π_2, then $g(s, x)$ is a local optimum for x.

PLS-reductions have standard properties. If Π_1 PLS-reduces to Π_2 and Π_2 PLS-reduces to Π_3 then Π_1 PLS-reduces to Π_3. Moreover, $\Pi_1 \in P_{\text{PLS}}$ if $\Pi_2 \in P_{\text{PLS}}$.

Lemma 1. *Let $\Pi_1 = (OP, N_1)$, $\Pi_2 = (OP, N_2)$ be two PLS problems and $N_1 \preceq N_2$. Then Π_1 PLS-reduces to Π_2.*

Proof. The proof is straightforward if we define the functions h and g as identical. □

We say that a problem Π in PLS is PLS-complete if every problem in PLS can be PLS-reduced to it. The following local search problems are PLS-complete (Yannakakis, 1997): The graph partitioning under the neighborhoods KL, Swap, FM, FM$_1$; Max-Cut problem under the Flip neighborhood and others.

Definition 5. Let Π be a local search problem and x be an instance of Π. The transition graph $\text{TG}_\Pi(x)$ of the instance x is a directed graph with one node for each feasible solution of x and with an arc $(s \to t)$ whenever $t \in N(s, x)$ and $F(t, x)$ is strictly better than $F(s, x)$ (i.e., greater if π is a maximization problem, and smaller if π is a minimization problem). The height of a node v is the length of the shortest path in $\text{TG}_\Pi(x)$ from v to a sink (a vertex with no outgoing arcs). The height of $\text{TG}_\Pi(x)$ is the largest height of a node.

The height of a node v is a lower bound on the number of iterations needed by the standard local descent algorithm even if it uses the best possible pivoting rule.

Definition 6. Let Π_1 and Π_2 be two local search problems, and let (h, g) be a PLS-reduction from Π_1 to Π_2. We say that the reduction is tight if for any instance x of Π_1 the height of $\text{TG}_{\Pi_2}(h(x))$ is at least as large as the height of $\text{TG}_{\Pi_1}(x)$.

It is clear that tight reductions compose. Tight reductions allow us to transfer lower bounds on the running time of the standard local search algorithm from one problem to another. Thus, if the standard algorithm of Π_1 takes exponential time in the worst case, then so does the standard algorithm for Π_2. Schäffer and Yannakakis (1991) prove the following sufficient condition for a PLS-reduction to be tight.

Lemma 2. *Suppose Π_1 and Π_2 are problems in PLS and let (h, g) be a PLS–reduction from Π_1 to Π_2. This reduction is tight if for any instance x of Π_1 there exists a subset R of feasible solutions for the image instance $h(x)$ such that the following properties hold*:

(1) *R contains all local optima of $h(x)$.*
(2) *For every solution p of x we can construct in polynomial time a solution $q \in R$ of $h(x)$ such that $g(q, x) = p$.*

(3) *Suppose that the transition graph of* $h(x)$, $\mathrm{TG}_{\Pi_2}(h(x))$ *contains a directed path from* $q \in R$ *to* $q' \in R$ *such that all internal path nodes are outside* R, *and let* $p = g(q,x)$ *and* $p' = g(q',x)$ *be the corresponding solutions of* x. *Then either* $p = p'$ *or* $TG_{\Pi_1}(x)$ *contains an arc from* p *to* p'.

Lemma 3. *Let* $\Pi_1 = (OP, N_1)$, $\Pi_2 = (OP, N_2)$ *be two PLS problems and* $N_1 \preceq N_2$. *Assume that for any instance* x *of OP the transition graph* $TG_{\Pi_2}(x)$ *is a subgraph of* $TG_{\Pi_1}(x)$. *Then* Π_1 *is tight PLS-reducible to* Π_2.

Proof. As for Lemma 1, the identical functions (h, g) define a PLS-reduction from Π_1 to Π_2. Now we show that this reduction is tight. Let x be an instance of Π_1. Since $N_1 \preceq N_2$ and the transition graph $\mathrm{TG}_{\Pi_2}(x)$ is a subgraph of $\mathrm{TG}_{\Pi_1}(x)$, it follows that the set of N_2 local optima is coincide with the set of N_1 local optima and the height of each node in $\mathrm{TG}_{\Pi_2}(x)$ is at least as large as the height of the node in $\mathrm{TG}_{\Pi_1}(x)$. Therefore, for any instance x of Π_1 the height of $\mathrm{TG}_{\Pi_2}(h(x))$ is at least as large as the height of $\mathrm{TG}_{\Pi_1}(x)$. □

5. The worst case complexity

Let us consider the graph partitioning problem.

Instance: Graph $G = (V, E)$ with *2n* nodes and a weight function $w : E \to Z$.
Solution: A partition of V into sets V_1, V_2 such that $|V_1| = |V_2| = n$.
Measure: The weight of the cut (V_1, V_2), i.e., the sum of the weights of the edges with one endpoint in V_I and another endpoint in V_2.
Goal: Max.

An FM_1 neighborhood for this problem is defined as FM_1 neighborhood for the p-median problem. We claim that the p-median problem with the FM_1 neighborhood is the most difficult local search problem in the class PLS.

Theorem 2. *The p-median problem with the* FM_1 *neighborhood is tight PLS-complete.*

Proof. Informally, for a given graph $G = (V, E)$ we create a matrix (g_{ij}) which has two rows and one column for each node of G. Moreover, the matrix (g_{ij}) has additional column for each edge of G. As a result, we have a one to one correspondence between feasible solutions of the graph partitioning problem and the p-median problem for $p = |V|/2$. We show that the weight of a cut (V_1, V_2) plus the value of the objective function for the p-median problem is a constant for pair of correspondent solutions. So, we get a tight reduction if put R as the set of all feasible solutions of the p-median problem.

Let E_i be the set of edges which are incident with the node $i \in V$. Put

$$W_i = \sum_{e \in E_i} w_e, \quad W = \sum_{e \in E} w_e, \quad I = \{1, \ldots, |V|\},$$
$$J = \{1, \ldots, |E| + |V|\}, \quad p = |V|/2.$$

To each $j = 1, \ldots, |E|$ we assign the edge $e \in E$ and put

$$g_{ij} = \begin{cases} 0, & \text{if } e = (i_1, i_2), (i = i_1) \vee (i = i_2), \\ 2w_e, & \text{otherwise.} \end{cases}$$

To each $j = |E| + 1, \ldots, |E| + |V|$ we put

$$g_{ij} = \begin{cases} 0, & \text{if } i = j - |E|, \\ W - W_i, & \text{otherwise.} \end{cases}$$

For the cut (V_1, V_2) we put $S = V_1$. The proof of the theorem is based on the following equality:

$$\sum_{j\in J} \min_{i\in S} g_{ij} + W(V_1, V_2) = nW.$$

By definition we have

$$\sum_{j=1}^{|E|} \min_{i\in S} g_{ij} = 2\sum(w_e | e = (i_1, i_2), i_1, i_2 \notin S)$$

and

$$\sum_{j=1+|E|}^{|J|} \min_{i\in S} g_{ij} = \sum_{i\notin S}(W - W_i) = nW - \sum_{i\notin S} W_i.$$

Note that

$$\sum_{i\notin S} W_i = W(V_1, V_2) + \sum_{j=1}^{|E|} \min_{i\in S} g_{ij},$$

as desired. □

Corollary 1. *The local search problems for the p-median under the Swap, KL, KL_1, FM neighborhoods are tight PLS-complete.*

This statement follows from the tight PLS-completeness of the Graph Partitioning problem with Swap, KL, FM neighborhoods (Johnson et al., 1988; Yannakakis, 1997). Property Swap $\preceq$ KL_1 and Lemma 3 give us the rest of the statement.

Let $\Pi \in PLS$. *The standard local optimum problem* for Π is the following. We are given an instance of Π and an initial solution. The goal is to find a local optimum with respect to the neighborhood that would be produced by the standard local descent algorithm starting from the initial solution. It is known that there is a local search problem in the class PLS where standard local optimum problem is *PSPACE*-complete (Yannakakis, 1997). Moreover, if there is a tight PLS–reduction from a local search problem Π_1 to a problem Π_2, then there is a polynomial time reduction from the standard local optimum problem for Π_1 to the standard local optimum problem for Π_2. Combining these facts, Theorem 2, and Corollary 1 we obtain the following statement.

Corollary 2. *Standard local optimum problems for the p-median under Swap, KL, KL_1, FM, FM_1 neighborhoods are PSPACE–complete.*

Combining Lemmas 1 and 2 and Theorem 2 we obtain the following.

Corollary 3. *Suppose that the neighborhood N is stronger than the neighborhood FM_1 and the local search problem (p-median, N) belongs to the class PLS. Then (p-median, N) is PLS-complete.*

Corollary 4. *If $P_{\text{PLS}} \neq PLS$ and the local search problem (p-median, N) belongs to the class P_{PLS} then $FM_1 \not\preceq N$.*

It is known that there is a local search problem in the class PLS such that the standard local descent algorithm takes an exponential number of iterations (Yannakakis, 1997). Combining this fact with Theorem 2 and Corollary 1, we obtain the same property for the p-median problem.

Corollary 5. *The standard local descent algorithm takes an exponential number of iterations in the worst case for the local search problems p-median under the Swap, KL, KL_1, FM, FM_1 neighborhoods regardless of the tie-breaking and pivoting rules used.*

Now we present a family of instances and initial solutions for the p-median problem for which the local descent algorithm spends an exponential number of iterations to find an KL_1-optimal solution. To this end, we show a tight PLS-reduction of the Generalized Graph 2-Coloring problem (2-GGCP) with Flip neighborhood to (p-median, KL_1), and use the family of instances for (2-GGCP, Flip) with desired properties (Vredeveld and Lenstra, 2003). The 2-GGCP problem is the following.

Instance: A graph $G = (V, E)$ and a weight function $w : E \to Z$.
Solution: A color assignment $c : V \to \{1, 2\}$.
Measure: The sum of weights of the edges that have endpoints with the same color (monochromatic edges).
Goal: Min.

Given a solution, a Flip neighbor is obtained by choosing a vertex and assigning a new color. A solution is Flip-optimal if flipping any single vertex does not decrease the total weight of monochromatic edges. The (2-GGCP) problem with Flip neighborhood is tight PLS-complete (Vredeveld and Lenstra, 2003).

Theorem 3. *The local search problem* (2-*GGCP*, *Flip*) *is tightly PLS-reduced to the local search problem* (*p-median*, KL_1).

Proof. We put $I = \{1, \dots, 2|V|\}$, $J = \{1, \dots, |V| + 2|E|\}$, $p = |V|$, $W = \sum_{e\in E}|w_e| + 1$. For each vertex $v \in V$ we introduce two rows i_v, i'_v and a column j_v of matrix g_{ij}. For each edge $e = (j_1, j_2) \in E$, we introduce two columns $j_1(e)$, $j_2(e) \in J$. Put

$$g_{ij_v} = \begin{cases} 0 & \text{if } (i = i_v) \vee (i = i'_v), \\ W & \text{otherwise}, \end{cases} \quad i \in I, \ j_v = 1, \dots, |V|.$$

For $w_e \geqslant 0$ we define

$$g_{ij_1(e)} = \begin{cases} 0 & \text{if } (i = j_1) \vee (i = j_2), \\ w_e & \text{if } (i = j'_1) \vee (i = j'_2), \\ W & \text{otherwise}, \end{cases} \quad i \in I, \ e \in E,$$

$$g_{ij_2(e)} = \begin{cases} 0 & \text{if } (i = j'_1) \vee (i = j'_2), \\ w_e & \text{if } (i = j_1) \vee (i = j_2), \\ W & \text{otherwise}, \end{cases} \quad i \in I, \ e \in E.$$

For $w_e < 0$ we define

$$g_{ij_1(e)} = \begin{cases} w_e/2 & \text{if } (i = j_1) \vee (i = j'_2), \\ -w_e/2 & \text{if } (i = j'_1) \vee (i = j_2), \\ W & \text{otherwise}, \end{cases} \quad i \in I, \ e \in E,$$

$$g_{ij_2(e)} = \begin{cases} w_e/2 & \text{if } (i = j'_1) \vee (i = j_2), \\ -w_e/2 & \text{if } (i = j_1) \vee (i = j'_2), \\ W & \text{otherwise}, \end{cases} \quad i \in I, \ e \in E.$$

Fig. 1 shows the structure of matrix g_{ij}.

This reduction is polynomial. It maps the instances of the 2-GGCP problem into the set of instances for the p-median problem. Let $S \subset I$ be a KL$_1$-minimum . We claim that exactly one row i_v or i'_v belongs to S for each $v \in V$.

Assume that $i_v, i'_v \notin S$. By definition, $|S| = p = |V|$. Hence, there is a vertex $v_0 \in V$ such that $i_{v_0}, i'_{v_0} \in S$. Let us consider a new solution $\widetilde{S} = (S \setminus \{i_{v_0}\}) \cup \{i'_v\}$. Obviously, $F(\widetilde{S}) < F(S)$ and S is not KL$_1$-minimum. The case $i_v, i'_v \in S$ is similar.

For the solution S we define a coloring assignment for the graph vertices:

$$c_S(v) = \begin{cases} 1 & \text{if } i_v \in S, \\ 2 & \text{otherwise}. \end{cases}$$

We wish to check that $c_S(v)$ is a Flip-minimum if and only if S is KL$_1$-minimum. Moreover, the objective values for these solutions are the same.

For each edge $e = (j_1(e), j_2(e))$ there are two rows j_1 and j_2 which correspond to the vertices $j_1(e)$ and $j_2(e)$. We claim that

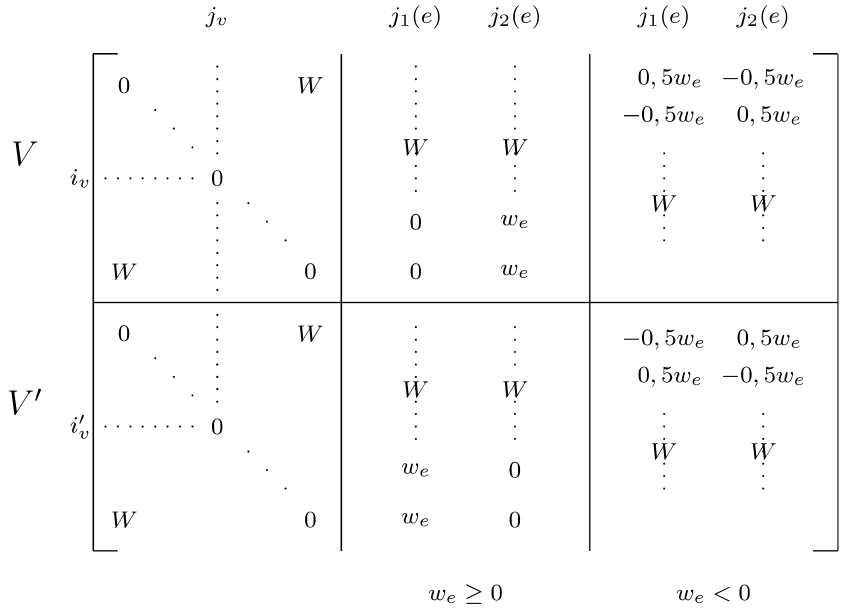

Fig. 1. The structure of matrix g_{ij}.

$$\min_{i \in S} g_{ij_1(e)} + \min_{i \in S} g_{ij_2(e)} = w_e.$$

Suppose that an edge $e \in E$ is monochromatic. We consider two cases.

1. Case $w_e \geqslant 0$. Assume that $j_1, j_2 \in S$. So, $j'_1, j'_2 \notin S$ and

$$\min_{i \in S} g_{ij_1(e)} = 0, \quad \min_{i \in S} g_{ij_2(e)} = w_e.$$

Similarly, if $j_1, j_2 \notin S$ then $j'_1, j'_2 \in S$ and

$$\min_{i \in S} g_{ij_1(e)} = w_e, \quad \min_{i \in S} g_{ij_2(e)} = 0.$$

2. Case $w_e < 0$. If $j_1, j_2 \in S$ then $j'_1, j'_2 \notin S$ and

$$\min_{i \in S} g_{ij_1(e)} = w_e/2, \quad \min_{i \in S} g_{ij_2(e)} = w_e/2.$$

Similarly, if $j_1, j_2 \notin S$ then $j'_1, j'_2 \in S$ and

$$\min_{i \in S} g_{ij_1(e)} = w_e/2, \quad \min_{i \in S} g_{ij_2(e)} = w_e/2.$$

Therefore, in both cases the equation holds and w_e is a part of the objective value $F(S)$.

Let us consider an edge $e \in E$ that has end points with different colors. Now we have either $j_1, j'_2 \in S$ or $j'_1, j_2 \in S$.

1. Case $w_e \geqslant 0$. If $j_1, j'_2 \in S$ then $j'_1, j_2 \notin S$ and

$$\min_{i \in S} g_{ij_1(e)} = 0, \quad \min_{i \in S} g_{ij_2(e)} = 0.$$

Similarly, if $j'_1, j_2 \in S$ then $j_1, j'_2 \notin S$ and

$$\min_{i \in S} g_{ij_1(e)} = 0, \quad \min_{i \in S} g_{ij_2(e)} = 0.$$

2. Case $w_e < 0$. If $j'_1, j_2 \in S$ then $j_1, j'_2 \notin S$ and

$$\min_{i \in S} g_{ij_1(e)} = -w_e/2, \quad \min_{i \in S} g_{ij_2(e)} = w_e/2.$$

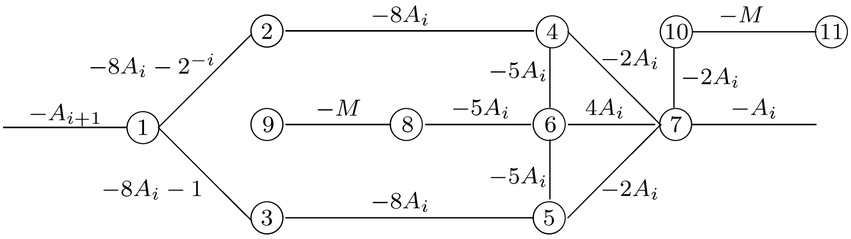

Fig. 2. Module $i : A_i = 20^{i-1}$.

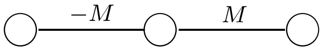

Fig. 3. Chain with large weights.

Similarly, if $j_1, j_2' \in S$ then $j_1', j_2 \notin S$ and

$$\min_{i\in S} g_{ij_1(e)} = w_e/2, \quad \min_{i\in S} g_{ij_2(e)} = -w_e/2.$$

Hence, the value w_e is not included into the objective value $F(S)$. In other words, our reduction saves the values of the objective functions for local minima. We now verify that it is tight PLS-reduction.

Let us assume that S is a KL_1-minimum but the corresponding color assignment $c_S(v)$ is not Flip-minimum. For this case we can find a vertex $v \in V$ and change the color of v with decreasing the total weight of monochromatic edges. But this transformation corresponds to swapping i_v and $i_{v'}$ for the solution S with the same decreasing of the objective function. Hence, S is not KL_1-minimum. A contradiction. So, we have a PLS-reduction.

Let R be the set of feasible solutions in which either $i_v \in S$ or $i_v' \in S$ for all $v \in V$. If $i_v \in S$ then $c_S(v) = 1$, otherwise $c_S(v) = 2$. It is a one to one correspondence between the elements of set R and color assignments. This choice of R satisfies Conditions 1 and 2 of Lemma 2. Suppose that the transition graph of the local search problem (p-median, KL_1), contains a directed path from $S \in R$ to $S' \in R$, such that all internal path nodes are outside R. Let S'' is a internal path node. Thus we have $F(S) > F(S'') > F(S')$. Since $S'' \notin R$, it follows that $F(S) < F(S'')$. Hence, each directed path with endpoints in R belongs to R. But each arc (S, S'), where $S, S' \in R$, corresponds to an arc $(c_S, c_{S'})$ in corresponding transition graph of 2-GGCP problem. So, we have Condition 3 of Lemma 2 satisfied, and our reduction is tight. □

We now describe an example for the 2-GGCP where local descent algorithm spends an exponential number of iterations to reach a Flip-optimum if it uses the best improvement pivoting rule. Graph $G = (V, E)$ for this example consists of K modules and a chain of three vertices as shown in Figs. 2 and 3.

Each module consists of 11 vertices. Vertex 1 is called the input node of the module. Vertex 7 is called the output node of the module. The input node of module i is adjacent to the output node of module $i+1$, for $i = K-1, \ldots, 1$. The input node of module K is adjacent to the right most vertex of the chain. Each edge has a weight. The large positive weight M makes sure that the two vertices incident to an edge have different colors for every Flip-optimum. It is known (Vredeveld and Lenstra, 2003) that the local descent algorithm with the best improvement pivoting rule flips the output node of the first module 2^K times if it starts from an initial solution where all vertices have the same color. We showed a tight PLS-reduction of this local search problem to (p-median, KL_1). Hence, we have a correspondent example for the (p-median, KL_1) as well.

6. Approximate local search

For any $\varepsilon > 0$ a solution S^ε is called an (ε, N)-local minimum if $F(S^\varepsilon) \leqslant (1+\varepsilon)F(S)$ for all $S \in N(S^\varepsilon)$. We show that an (ε, N)-local minimum can be found for the p-median problem in polynomial time both in the problem size and $1/\varepsilon$. In fact, we will show the existence of a fully polynomial time ε-local optimization scheme for the p-median problem with a polynomially searchable neighborhood.

Property 1. *If a neighborhood N is polynomially searchable then an (ε, N)-local minimum for the p-median problem can be found in polynomial time both in the problem size and $1/\varepsilon$.*

In order to get the desired scheme we apply the approach of Orlin et al. (2004) for arbitrary 0–1 linear programming problems. Let us modify the matrix (g_{ij}) by scaling each element, $g'_{ij} = \lceil g_{ij}/\theta \rceil \theta$ by an appropriate multiple $\theta > 0$ and use the standard local descent algorithm with an arbitrary starting solution S^0. Assume that we have got a local minimum S^0 for this new objective function $F'(S) = \sum_{j \in J} \min_{i \in S} g'_{ij}$ and there exists a constant Δ such that $m\theta < \Delta \leqslant F'(S^0) \leqslant F'(S^0)$. In this case we have

$$F(S^0) \leqslant F'(S^0) \leqslant F'(S) \leqslant F(S) + m\theta \quad \text{for all } S \in N(S^0).$$

Hence,

$$F(S^0) \leqslant F(S)(1 + m\theta/F(S)) \leqslant F(S)(1 + m\theta/(F(S^0 - m\theta))) \leqslant F(S)(1 + m\theta/(\Delta - m\theta))$$

and we get an (ε, N)-local minimum if $\theta = \varepsilon\Delta/(m(\varepsilon + 1))$. But we cannot guarantee that the number of steps is polynomial for this simple algorithm. It is pseudo-polynomial algorithm only. For each step of local descent, the objective value decreases as least by θ. So, the number of steps is $\mathrm{O}(F'(S^0)/\theta) = \mathrm{O}(m\varepsilon^{-1}F'(S^0)/\Delta)$.

To get rid of this problem, we modify our algorithm as follows. Put $\Delta = F(S^0)/2,\ \theta = \varepsilon\Delta/(m(\varepsilon + 1))$, and apply the local descent until $F(S) < \Delta$ or S is a local minimum for the modified objective function. If we reach a local minimum then algorithm stops and returns the current solution S as an (ε, N)-local minimum for the original problem. Otherwise, put $S^0 = S,\ \Delta = F(S^0)/2,\ \theta = \varepsilon\Delta/(m(\varepsilon + 1))$ and repeat the local descent again. We will change Δ at most $\mathrm{O}(\log F(S^0))$ times and spend at most $\mathrm{O}(m\varepsilon^{-1})$ steps for any Δ. So, we have got a fully polynomial time ε-local optimization scheme for the p-median problem with a polynomially searchable neighborhood.

Property 2. *If $FM_1 \preceq N$ and there is a polynomial time algorithm to find a feasible solution S^0 for the p-median problem such that $F(S^0) \leqslant F(S) + 2^{q(n,m)}$ for any fixed polynomial q(n,m) and all $S \in N(S^0)$, then one can find a local optimum in polynomial time for all problems in the class PLS.*

In other words, if $P_{\mathrm{PLS}} \neq PLS$ then we cannot guarantee any amount of absolute deviation of the local optimum in polynomial time. To confirm this claim, we consider a new instance $g'_{ij} = g_{ij}(1 + 2^{q(n,m)})$ and apply the algorithm to it. Let S_A be a solution returned by the algorithm. Without loss of generality, we may assume that all elements of the matrix (g_{ij}) are integers. Solution S_A is feasible for the original problem, and $F(S_A) - F(S) \leqslant 2^{q(n,m)}/(1 + 2^{q(n,m)}) < 1$ for all $S \in N(S_A)$. Hence, S_A is an N-optimal solution.

7. Pivoting rules

Let $\mathrm{Swap}^*(S) = \{S' \in \mathrm{Swap}(S) | F(S') < F(S)\}$ be the subset of neighbors for S with better values of the objective function than S. The pivoting rule selects a neighbor for the current solution at each step of the local descent. This choice may affect the complexity of the algorithm drastically. We consider six pivoting rules and analyse their influence on the number of steps and relative error of the local optima obtained. Some of these rules are well known and used in metaheuristics. The others are new and help us to understand the properties of the corresponding transition graph better.

The Best improvement rule selects a solution in the set $\mathrm{Swap}^*(S)$ with the smallest value of the objective function. If there are several best elements we pick up the lexicographical minimal one. It seems that this rule is the most popular in the local search methods (Resender and Werneck, 2003).

The Worst improvement rule selects a solution in $\mathrm{Swap}^*(S)$ with the largest value of the objective function. According to this rule we use the most flat direction for descent. So, we may guess that this rule produces more steps and the final local minimum may be better than for the previous case.

The Random improvement rule picks a neighbor for S in the set $\mathrm{Swap}^*(S)$ at random with uniform distribution. It is one of the fastest pivoting rule, and can lead to different local optima from the same starting solution.

The First improvement rule is one of the famous pivoting rules. It prescribes to use an element from $\mathrm{Swap}^*(S)$ which is found in $\mathrm{Swap}(S)$ first. We test the neighbors of S in the lexicographical order and terminate when the first better neighbor is discovered.

The Circular rule is closely related to the previous one. It differs from it in one point only. The *First improvement* rule begins the search at every step from the same starting position, for example, from the lexicographically minimal position. The *Circular* rule begins from the position where the previous step terminates (Papadimitriou and Steiglitz, 1982). The idea of this rule is based on the following observation. In many cases, the unprofitable moves for the current solution will be unprofitable for the neighboring solutions. So, it is better to continue exploring instead of starting from the initial position.

Finally, the rule of maximal *Freedom* selects a neighbor S' in the set Swap$^*(S)$ with the maximum cardinality of the set Swap$^*(S')$. This rule is more time consuming but gives us a neighbor with the maximum number of directions for further improvement. The number of elements in the set Swap$^*(S)$ is called the freedom of solution S.

8. Computational experiments

We test the local descent algorithm with described pivoting rules on random instances. For all instances, we put $n = m$. The values g_{ij} are taken from interval $[0, 1000]$ at random with the uniform distribution. We generate 30 instances and study two cases, $p = [n/10]$ and $p = 15$. The goal of our experiments is to investigate the influence of the pivoting rules on the number of steps of the local descent algorithm and compare the relative deviations of the local optima obtained.

Figs. 4 and 5 show the average number of steps from random starting solution to a Swap-optimum, $p = [n/10]$. Every point at the curve is the average value for 100 trials. The pivoting rule *Freedom* is presented at both figures. For all rules except *Worst*, the number of steps grows as a linear function of n. For the *Worst* rule we see a superlinear function. The number of steps for local descent grows rapidly, and the difference between the *Worst* and the *Best* rules becomes extremely high for $n > 100$. So, pivoting rules are important from the viewpoint of running time. Fig. 6 confirms the conclusion for the relative error as well. The *Best* rule has a large average deviation from the best solution found. The *Freedom* rule shows the smallest deviation. We

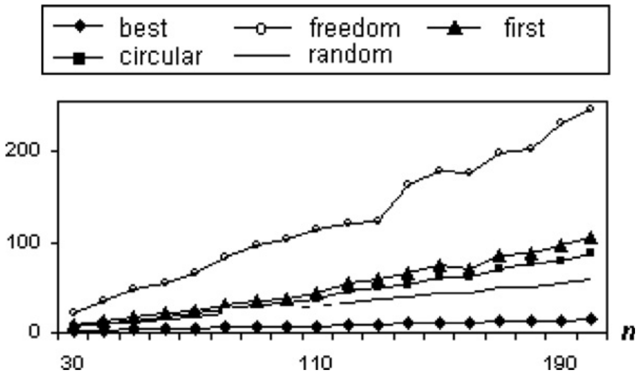

Fig. 4. The average number of steps without worst rule, $p = n/10$.

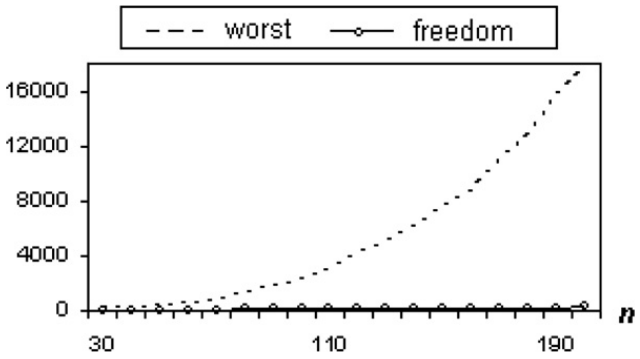

Fig. 5. The average number of steps for worst and freedom rules, $p = n/10$.

believe this rule tends to find a local optimum with a large basin of attraction, and this is a reason why we get high quality local optima.

Figs. 7 and 8 illustrate the average number of steps for the case $p = 15$. All rules show linear functions for the average number of steps. The *Worst* rule has the largest number of steps but its relative error is close to the rules *First*, *Circular*, and *Random*. The *Best* rule leads to the local minima with large relative errors (see Fig. 9). The same behavior of the local descent algorithm we have observed for Euclidean instances when the elements g_{ij} are Euclidean distances for random points on the two dimensional plane.

It is known (Tovey, 1997) that the local descent algorithm is polynomial on average for random functions on the 0–1 hypercube with polynomial neighborhoods. Similar results we can obtained for p-layer of the hypercube and the Swap neighborhood. More precisely, let $F(S)$ be a random function and for each $S \subset I,\ |S| = p$, the value $F(S)$ is selected independently with given probability distribution. If p is a constant

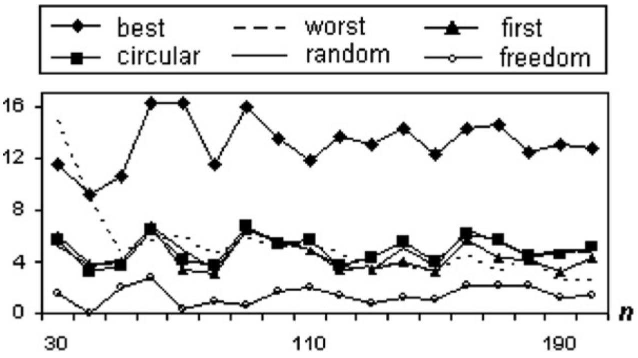

Fig. 6. The average relative error (%), $p = n/10$.

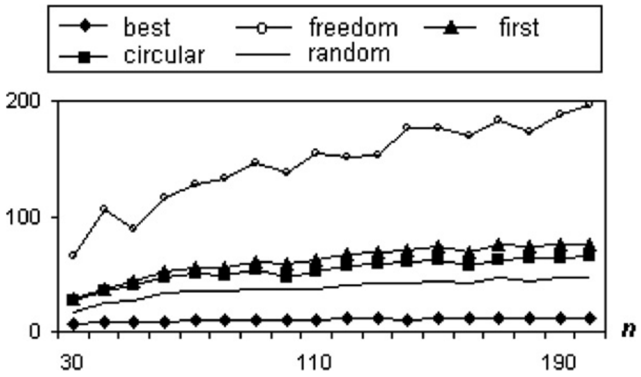

Fig. 7. The average number of steps without worst rule, $p = 15$.

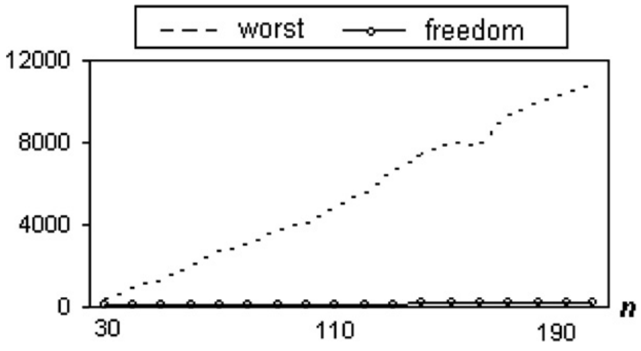

Fig. 8. The average number of steps for worst and freedom rules, $p = 15$.

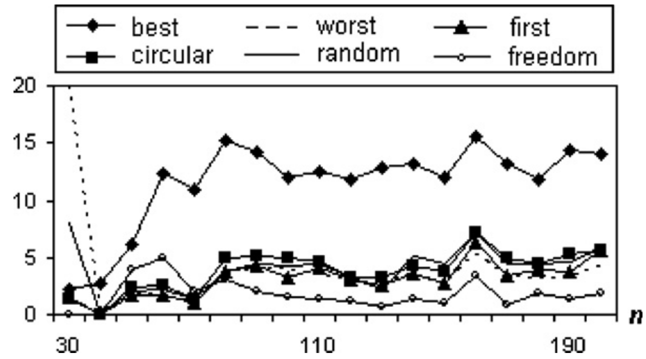

Fig. 9. The average relative error (%), $p = 15$.

then the expected number of steps for the standard local descent algorithm with Swap neighborhood is less than $1,5epn$ regardless of the tie-breaking and pivoting rules used, where e is logarithmic constant. Our computational results show the same behavior of the local descent algorithm for random matrices. If $p = [\alpha n]$ for given $0 < \alpha < 1$ then the expected number of steps for the standard local descent algorithm with Swap neighborhood for random function $F(S)$ is less than $1,5en^2$ regardless of the tie-breaking and pivoting rules used. Our computational results for the p-median problem show the linear function for all pivoting rules except the *Worst*. For this rule we have a nonlinear function. It is interesting to study theoretically the behavior of the local descent algorithm for random matrices g_{ij} as well, not only for random functions $F(S)$ for the p-layer of the hypercube.

9. Conclusions

For the p-median problem, we shown that the standard local descent algorithm takes an exponential number of steps in the worst case. We introduced several neighborhoods and proved that the corresponding local search problems are tightly PLS-complete. We illustrated the relationship between the Swap local optima, classical Karush–Kuhn–Tucker conditions, and 0–1 local saddle points.

In further research, it may be interesting to study the distribution of local optima in the feasible domain and understand the complexity of local search problems for Euclidean matrices. The metric case is very important for theoretical research. Some approximate algorithms with guaranteed performance ratio are based on the local descent with Swap and k-Swap neighborhoods (Arya et al., 2004; Korte and Vygen, 2005). But the number of steps to reach local optimum could be exponential. Still it is not clear whether this metric case is polynomially solvable or PLS-complete.

Acknowledgements

The authors thank Sergey Avgustinovich for fruitful discussions and useful remarks. This research was supported by Russian Foundation for Basic Research, grant 06-01-00075.

References

Arya, V., Garg, N., Khandekar, R., Meyerson, A., Munagala, K., Pandit, V., 2004. Local search heuristics for k-median and facility location problems. SIAM Journal on Computing 33, 544–562.

Ausiello, G., Gambosi, P., Crescenzi, G., Kann, V., Marchetti-Spaccamela, A., Protasi, M., 1999. Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties. Springer, Berlin.

Beresnev, V.L., Gimadi, E.Kh., Dementèv, V.T., 1978. Extremal Standardization Problems. Nauka, Novosibirsk (in Russian).

Fiduccia, C.M., Mattheyses, R.M., 1982. A linear time heuristic for improving network partitions. In: ACM IEEE Nineteenth Design Automation Conference: Proceedings. IEEE Computer Society Press, pp. 175–181.

Johnson, D.C., Papadimitriou, C.H., Yannakakis, M., 1988. How easy is local search? Journal of Computer and system Sciences 37 (1), 56–87.

Kernighan, B.W., Lin, S., 1970. An effective heuristic procedure for partitioning graphs. The Bell System Technical Journal 49, 291–307.

Korte, B., Vygen, J., 2005. Combinatorial Optimization. Theory and Algorithms, third ed. Springer.

Mladenović, N., Brimberg, J., Hansen, P., Moreno-Pérez, J., 2007. The *p*-median problem: A survey of metaheuristic approaches. European Journal of Operational Research 179 (3), 927–939.

Nemhauser, G.L., Wolsey, L.A., 1988. Integer and Combinatorial Optimization. John Wiley & Sons.

Orlin, J., Punnen, A., Schulz, A., 2004. Approximate local search in combinatorial optimization. In: Proceedings of the 15th Annual ACM–SIAM Symposium on Discrete Algorithms, New Orleans, Louisiana.

Papadimitriou, C., Steiglitz, K., 1982. Combinatorial Optimization: Algorithms and Complexity. Prentice-Hall.

Resender, M., Werneck, R.F., 2003. On the implementation of a swap-based local search procedure for the *p*-median problem. In: Ladner, R.E. (Ed.), Proceedings of the Fifth Workshop on Algorithm Engineering and Experiments. SIAM, pp. 119–127.

Schäffer, A.A., Yannakakis, M., 1991. Simple local search problems that are hard to solve. SIAM Journal on Computing 20 (1), 56–87.

Tovey, C.A., 1997. Local improvement on discrete structures. In: Aarts, E., Lenstra, J.K. (Eds.), Local Search in Combinatorial Optimization. John Wiley & Sons, Chichester, pp. 57–89.

Vredeveld, T., Lenstra, J.K., 2003. On local search for the generalized graph coloring problem. Operations Research Letters 31 (4), 28–34.

Yannakakis, M., 1997. Computational complexity. In: Aarts, E., Lenstra, J.K. (Eds.), Local Search in Combinatorial Optimization. John Wiley & Sons, Chichester, pp. 9–55.