J Comb Optim (2015) 29:451–471
DOI 10.1007/s10878-013-9604-1

Embedding signed graphs in the line

Heuristics to solve MinSA problem

Eduardo G. Pardo · Mauricio Soto · Christopher Thraves

Published online: 28 March 2013

Abstract Signed graphs are graphs with an assignment of a positive or a negative sign to each edge. These graphs are helpful to represent different types of networks. For instance, they have been used in social networks, where a positive sign in an edge represents friendship between the two endpoints of that edge, while a negative sign represents enmity. Given a signed graph, an important question is how to embed such a graph in a metric space so that in the embedding every vertex is closer to its positive neighbors than to its negative ones. This problem is known as Sitting Arrangement (SA) problem and it was introduced by Kermarrec et al. (Proceedings of the 36th International Symposium on Mathematical Foundations of Computer Science (MFCS), pp. 388–399, 2011). Cygan et al. (Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS), 2012) proved that the decision version of SA problem is NP-Complete when the signed graph has to be embedded into the Euclidean line. In this work, we study the minimization version of SA (MinSA) problem in the Euclidean line. We relate MinSA problem to the well known quadratic assignment (QA) problem. We establish such a relation by proving that local minimums in MinSA problem are equivalent to local minimums in a particular case of QA problem. In this document, we design two heuristics based on the combinatorial

E. G. Pardo
Universidad Rey Juan Carlos, Office 131, Dept-II, C/Tulipán s/n, 28933 Móstoles, Madrid, Spain
e-mail: eduardo.pardo@urjc.es

M. Soto
LIFO, Université d'Orléans, Bat. 3IA, Rue Léonard de Vinci, B.P. 6759, 45067 Orleans Cedex 2, France
e-mail: mauricio.soto@univ-orleans.fr

C. Thraves (✉)
Universidad Rey Juan Carlos, Office 101, Dept-III, Camino del Molino s/n, 28943 Fuenlabrada, Madrid, Spain
e-mail: cbthraves@gsyc.es

structure of MinSA problem. We experimentally compare their performances against heuristics designed for QA problem. This comparison favors the proposed heuristics.

Keywords Signed graphs · Graph embedding · Graph drawing · Heuristics · Quadratic assignment problem

1 Introduction

A signed graph is an undirected graph $G = (V, E)$ together with a function $f : E \to \{+1, -1\}$ that assigns either a positive or a negative sign to each edge. Signed graphs have a tremendous utility at the moment of representing *opposite relations* between two vertices in a network. For instance, *friendship* and *enmity* in social networks are opposite relations modeled appropriately via signed graphs, as well as the requirement of *existence* or *absence* of a communication link between two vertices in a communication network.

Some networks impose geometric requirements in the deployment of the vertices. In the case of radio networks, for instance, the communication is carried out by means of a radio channel where vertices may broadcast messages to all neighboring vertices (vertices deployed within communication range). Therefore, if beforehand two vertices need to have a direct communication link between them, in the deployment they have to be placed nearby. While, if the same two vertices must be disconnected (i.e., a direct communication link between them is forbidden), they have to be deployed afar. Similarly, in a social context when guests have to be sit in a dinner table, if beforehand the host knows that when two guests are sitting together they produce a pleasant environment, they should be sit close at hand. While, in the opposite case, they should be sit far apart.

In this way, *sitting arrangement* (SA) problem appears naturally: *Given a signed graph* $G = (V, E)$ *and a metric space* $\mathcal{M}$, *does exist an embedding* $D : V \to \mathcal{M}$ *such that for every pair of incident edges* $(x, y), (x, w)$ *with* $f((x, y)) = +1$ and $f((x, w)) = -1$, *the embedding satisfies* $d(D(x), D(y)) < d(D(x), D(w))$? *where* d *is the distance in the metric space* $\mathcal{M}$. This problem captures the idea that each vertex has to be placed closer to its positive neighbors than to its negative neighbors.

A.-M. Kermarrec et al. introduced SA problem in Kermarrec and Thraves (2011). In their work, the authors proved that when the input signed graph is complete[1], and the metric space is the Euclidean line, it is decidable in polynomial time whether such an embedding exists or not. Furthermore, in the case when the input admits an embedding, they proposed an algorithm that computes one in polynomial time. Nevertheless, Cygan et al. proved in Cygan et al. (2012) that when the input is not restricted to complete signed graphs and the metric space is the Euclidean line, SA problem is NP-Complete.

The last mentioned result motivates us to search for approximate solutions for the optimization version of the SA problem in the Euclidean line (*MinSA*). The goal of

[1] A signed graph is complete if every pair of vertices is connected by a signed edge.

MinSA problem is to minimize the number of *errors* produced in the embedding, where an *error is a pair of incident edges* (x, y), (x, w) *such that* $(x, y) \in E^+$, $(x, w) \in E^-$ *and* $|D(x) - D(y)| \geq |D(x) - D(w)|$.

1.1 Related work

Different problems have been studied using signed graphs. Bansal et al. (2004) studied the clustering correlation problem on signed graphs. Antal et al. (2005) studied the dynamic of social networks represented via signed graphs. The structure of on-line social networks represented via signed graphs is also studied by Szell et al. (2010) and Leskovec et al. (2010b). Prediction of signed links based on balance and status theory is studied by Leskovec et al. (2010a).

More closely related, SA problem was previously studied by Kunegis et al. (2010) with the aim of clustering and visualization of signed graphs in the Euclidean plane. Motivated by the huge amount of social networks existent in the Internet that can be modeled via signed graphs, the authors of Kunegis et al. (2010) used spectral analysis to empirically compute visualizations of signed graphs, where the proposed visualization had the same goal as the one in SA problem. But, SA problem was not properly introduced until A.-M. Kermarrec et al. did it in Kermarrec and Thraves (2011). The authors of Kermarrec and Thraves (2011) introduced a general definition for SA problem, also they studied the connection between embedding signed graphs following their definition, balance theory (Harary 1953; Cartwright and Harary 1956; Harary et al. 1980) and clusterizable graphs (Davis 1967). They showed that the family of graphs that admits an embedding in the Euclidean line contains balanced signed graphs and clusterizable signed graphs. Hence, they claimed that social networks are better represented by this embedding than balance theory or clusterizable graphs. Kermarrec et al. also proved that when the input graph is a complete signed graph SA problem can be solved in polynomial time and presented an algorithm that computes the embedding when it is possible. Subsequently, Cygan et al. (2012) proved that SA problem is NP-complete when there is no restriction for the input and the host space is the Euclidean line. They also showed that the existence of a subexponential time algorithm would violate the Exponential Time Hypothesis. Finally, they presented a single-exponential time algorithm based on dynamic programming.

1.2 Our results

In this work, we first connect MinSA problem with the well known *quadratic assignment* (QA) problem. Though these optimization problems have different objective functions, we show that they are closely related. We show that, indeed, in the case when the input signed graph is complete, a local minimum for MinSA problem is equivalent to a local minimum for QA problem. This relation encourages the utilization of QA problem heuristics in our study. Additionally, we propose two heuristics to approximate MinSA problem. The first proposed heuristic is purely greedy, while the second one includes some randomization according to the ideas proposed in the

Greedy Randomized Adaptive Search (GRASP) methodology (Feo and Resende 1989, 1995). We experimentally compare these heuristics against two well known heuristics designed for QA problem. The experimental results show that the purely greedy heuristic found the optimal solution in most of the cases when its value is zero. In other words, the greedy heuristic shows to be capable to recognize, in the evaluated instances, whether a signed graph has an embedding with no errors in the Euclidean line. On the other hand, experimental results allow us to conclude that both heuristics provide more accurate solutions than those provided by heuristics designed for QA problem in less computing time.

The rest of the document is organized as follows: Sect. 2 contains the notation and a precise definition of the studied problem. In Sect. 3 the relationship between MinSA problem and QA problem is presented. Section 4 includes structural properties which help in the design of the heuristics. Section 5 presents the proposed heuristics to approximate the optimal solution of MinSA problem. In Sect. 6, we describe the experiments carried out and the obtained results. Finally, Sect. 7 presents our conclusions and future work.

2 Definitions

In order to represent a signed graph we use $G = (V, E^+ \cup E^-)$, where V is the set of vertices, and E^+ (resp., E^-) is the set of positive (respectively, negative) edges. We denote by n and m the size of V and the size of $E = E^+ \cup E^-$, respectively. Given a vertex $x \in V$, the set of neighbors of x is denoted by $N(x)$, while $d(x)$ denotes the degree of x. Also, we denote by $N^+(x)$ (respectively, $N^-(x)$) the set of positive (respectively, negative) neighbors of x and by $d^+(x)$ (respectively, $d^-(x)$) the positive (respectively, negative) degree of x.

The version of SA problem for a general metric space is described in the previous section. Nevertheless in this work, we are particularly interested in the case when the metric space $\mathcal{M}$ is the Euclidean line.

Definition 1 (*SA problem in the line*) Given a signed graph $G = (V, E^+ \cup E^-)$, is there an embedding $D : V \to \mathbb{R}$ such that for every pair of incident edges $(x, y), (x, w)$ with $(x, y) \in E^+$ and $(x, w) \in E^-$, such that the embedding satisfies: $|D(x) - D(y)| < |D(x) - D(w)|$?

On the other hand, let us define an ordering of the set of vertices V as an injection $\pi : V \to \{1, 2, 3, \ldots, n\}$. Kermarrec and Thraves (2011) proved that SA problem in the line can be restated as a combinatorial problem.

Theorem 1 (Lemma 3 and 4 in Kermarrec and Thraves *2011*) *There exists an embedding that satisfies Definition 1 for a given signed graph $G = (V, E^+ \cup E^-)$ if and only if there exists an ordering π of the set of vertices V such that:*

(i) $(y, x) \in E^-$ *and* $\pi(y) < \pi(x) \Rightarrow \forall\, w$ *such that* $\pi(w) < \pi(y)$, $(w, x) \notin E^+$,
(ii) $(y, x) \in E^-$ *and* $\pi(x) < \pi(y) \Rightarrow \forall\, w$ *such that* $\pi(y) < \pi(w)$, $(w, x) \notin E^+$.

Hence, finding an embedding of V in the Euclidean line that satisfies Definition 1 is equivalent to find an ordering of V that satisfies conditions (i) and (ii). Given the characteristics of condition (i) and (ii), if previously an error produced by an embedding D was understood as a pair of incident edges that violates the requirement of the embedding, now an error is defined in the following way.

Definition 2 (*Error in a vertex*) Given a signed graph $G = (V, E^+ \cup E^-)$ and an ordering π of V, an error in vertex x produced by π is defined as a pair of vertices y, w in V such that $(x, y) \in E^+$, $(x, w) \in E^-$ and $\pi(y) < \pi(w) < \pi(x)$ or $\pi(x) < \pi(w) < \pi(y)$.

As we have said, M. Cygan et al. recently proved in Cygan et al. (2012) that SA problem in the line is NP-complete when there is no restriction imposed to the input. In consequence, it is worth moving to the *minimization version of SA* (MinSA) problem in the line and attempt to approximate the solution.

Consequently, if we denote by $\mathcal{E}_x^\pi$ the number of errors in vertex x produced by an ordering π, *MinSA* problem in the Euclidean line is defined as follows:

Definition 3 (*MinSA problem*) Given a signed graph G, find an ordering π of the set of vertices V such that the total number of errors

$$\mathcal{E}(\pi) = \sum_{x \in V} \mathcal{E}_x^\pi$$

is minimized.

3 Relationship between MinSA and QA problem

In this section we show the relationship between MinSA and QA problem. First we define a quadratic formulation for MinSA problem, to then show that local minimums in both MinSA an its quadratic formulation are equivalent.

3.1 Quadratic assignment formulation

The QA problem is an important combinatorial optimization problem introduced by Koopmans and Beckmann (1957). Since then it has been widely studied, QA problem has been applied in different problems, e.g., matching, partitioning, TSP, facility location problems, process communication, and scheduling among others. Classic surveys on QA problem are Pardalos et al. (1994), Burkard et al. (1998) and Çela (1998). Sahni and Gonzalez (1976) proved that QA problem is NP-complete, even more, they show that any algorithm computing a ϵ-approximation is also NP-complete. On the other hand, numerous heuristics have been developed for the QA problem with multiples techniques. A non extensive list includes simulated annealing (Burkard and Rendl 1984; Connolly 1990), tabu search (Skorin-Kapov 1990; Taillard 1991), and ant system (Taillard 1998). A detailed review of exact methods and heuristics can be found in Drezner et al. (2005), Commander (2007), Nehi and Gelareh (2007) and Loiola et al. (2007).

The QA problem is defined as follows: given two sets of equal size, say V and $[n] = \{1, 2, \ldots, n\}$, a weight function $\omega : V \times V \to \mathbb{R}$, and a distance function

$\sigma : [n] \times [n] \to \mathbb{R}$, find an ordering $\pi : V \to [n]$ such that the cost function $\sum_{x,y} \omega(x, y)\sigma(\pi(x), \pi(y))$ is minimized.

In order to model MinSA problem as a particular case of QA problem, we define the weight between two vertices based on the sign of the edge that connects them:

$$\omega(x, y) = \begin{cases} 1, & \text{if } (x, y) \in E^+; \\ 0, & \text{if } (x, y) \notin E; \\ -1, & \text{if } (x, y) \in E^-. \end{cases}$$

On the other hand, the distance between two positions in an ordering is defined as the difference between them, $\sigma(i, j) = |i - j|$. Now given an ordering π, let us denote by $\mathcal{QA}(\pi)$ the objective function of the quadratic formulation of MinSA problem evaluated at π. Then, the objective function $\mathcal{QA}(\pi)$ is:

$$\mathcal{QA}(\pi) = \sum_{(x,y)\in E^+} |\pi(x) - \pi(y)| - \sum_{(x,y)\in E^-} |\pi(x) - \pi(y)|,$$

the sum of the lengths of the positive edges minus the sum of the lengths of the negative edges.

We emphasize in the fact that the quadratic formulation of MinSA problem is not equivalent to the original MinSA problem presented in Definition 3. Indeed, the objective functions are different. Nonetheless, there exists a strong relationship between these two formulations. In this document, we show the existence of this relationship via two ways, both in theory in this section as well as experimentally in Sect. 6.

3.2 Equivalence between local minimums

In order to understand the relationship between the objective function for QA problem and the one defined for MinSA problem we can say that, intuitively, when minimizing $\mathcal{QA}(\cdot)$ it is necessary to search for an ordering that keeps the positive edges with the shortest possible length and, at the same time, keeps the negative edges with the longest possible length. Therefore, the friends of a vertex are kept close while its vertices enemies are kept faraway.

We focus on the case when the input signed graph is complete. Now, given an ordering π, each vertex has positive and negative edges going to its right and left hand side. Let us denote by $d_L^+(x) = |\{y \in N^+(x) : \pi(y) < \pi(x)\}|$ the amount of positive edges incident to x and going to its left, and by $d_R^+(x) = |\{y \in N^+(x) : \pi(y) > \pi(x)\}|$ the amount of positive edges incident to x and going to its right hand side. Since vertices are placed in different positions, the length of positive edges incident to a vertex x is at least $\sum_{i=1}^{d_R^+(x)} i + \sum_{i=1}^{d_L^+(x)} i$, which is the case when the $d_R^+(x)$ neighbors are placed in the first $d_R^+(x)$ positions next to the right of x, and the $d_L^+(x)$ neighbors are placed in the first $d_L^+(x)$ positions at the left of x. Furthermore, every error produced by π in x increases this length by one. Therefore, we can state the following inequality for x:

$$\sum_{y \in N^+(x)} |\pi(y) - \pi(x)| = \sum_{i=1}^{d_R^+(x)} i + \sum_{i=1}^{d_L^+(x)} i + \mathcal{E}_x^\pi,$$

and then, summing over all possible x vertices we obtain:

$$\sum_{(x,y) \in E^+} |\pi(y) - \pi(x)| = \frac{1}{2} \sum_{x \in V} \left(\sum_{i=1}^{d_R^+(x)} i + \sum_{i=1}^{d_L^+(x)} i \right) + \frac{1}{2} \mathcal{E}(\pi), \tag{1}$$

where $\mathcal{E}(\pi)$ is the number of errors produced by π (the objective function of MinSA problem). We point out that the factor one over two comes from the fact that the length of an edge is measured from the two endpoints of that edge, hence it has to be divided by two.

Equivalently, we can say that the length of negative edges incident to a vertex x is at most $\sum_{i=1}^{\pi(x)-1} i - \sum_{i=1}^{d_L^+(x)} i + \sum_{i=1}^{n-\pi(x)} i - \sum_{i=1}^{d_R^+(x)} i - \mathcal{E}_x^\pi$, which is the length of every edge incident to x (going to the left and to right hand side of x) minus the length of the positive edges incident to x. Note that in this case every error produced by π in x decreases that total length by one. Equivalently than before, we obtain the following equality:

$$\sum_{y \in N^-(x)} |\pi(y) - \pi(x)| = \sum_{i=1}^{\pi(x)-1} i - \sum_{i=1}^{d_L^+(x)} i + \sum_{i=1}^{n-\pi(x)} i - \sum_{i=1}^{d_R^+(x)} i - \mathcal{E}_x^\pi,$$

and then, summing over all possible x we obtain,

$$\sum_{(x,y) \in E^-} |\pi(y) - \pi(x)| = \frac{1}{2} \sum_{x \in V} \left(\sum_{i=1}^{\pi(x)-1} i - \sum_{i=1}^{d_L^+(x)} i + \sum_{i=1}^{n-\pi(x)} i - \sum_{i=1}^{d_R^+(x)} i \right) - \frac{1}{2} \mathcal{E}(\pi). \tag{2}$$

Now, taking the difference between Eqs. (1) and (2), we obtain an expression that connects the objective function of MinSA problem and the one for its quadratic program:

$$\sum_{(x,y) \in E^+} |\pi(x) - \pi(y)| - \sum_{(x,y) \in E^-} |\pi(x) - \pi(y)| = \frac{1}{2} \sum_{x \in V} \left(\sum_{i=1}^{d_R^+(x)} i + \sum_{i=1}^{d_L^+(x)} i \right) + \frac{1}{2} \mathcal{E}(\pi)$$

$$-\frac{1}{2}\sum_{x\in V}\left(\sum_{i=1}^{\pi(x)-1} i-\sum_{i=1}^{d_L^+(x)} i+\sum_{i=1}^{n-\pi(x)} i-\sum_{i=1}^{d_R^+(x)} i\right)+\frac{1}{2}\mathcal{E}(\pi)$$

$$=\sum_{x\in V}\left(\sum_{i=1}^{d_R^+(x)} i+\sum_{i=1}^{d_L^+(x)} i\right)+\mathcal{E}(\pi)-\frac{(n-1)n(n+1)}{6}$$

$$\mathcal{QA}(\pi)=\mathcal{E}(\pi)+\sum_{x\in V}\left(\sum_{i=1}^{d_R^+(x)} i+\sum_{i=1}^{d_L^+(x)} i\right)-\frac{(n-1)n(n+1)}{6}. \tag{3}$$

Going further, we can say that finding a local minimum is equivalent in both objective functions $\mathcal{QA}(\cdot)$ and $\mathcal{E}(\cdot)$. In order to prove that, we first define *locality* for an ordering. Given an ordering π, we say that two vertices y and x are *consecutive* in π if and only if $\pi(x)=\pi(y)+1$. Let us define now an ordering π' to be in the *locality* of an ordering π if and only if π' can be obtained from π by a single exchange of two consecutive vertices in π, i.e., $\pi'(w)=\pi(w)$ for all $w\in V$ except for two consecutive vertices y and x for which $\pi'(x)=\pi(y)$ and $\pi'(y)=\pi(x)$. *We say that an ordering* π *is a local minimum for* $\mathcal{QA}(\cdot)$ *(respectively, for* $\mathcal{E}(\cdot)$*) if and only if* $\mathcal{QA}(\pi)$ *(respectively* $\mathcal{E}(\pi)$*) is the minimum among* π*'s locality.*

Lemma 1 *When the input signed graph is complete, an ordering* π *is a local minimum for the objective function* $\mathcal{QA}(\cdot)$ *if and only if* π *is a local minimum for the objective function* $\mathcal{E}(\cdot)$.

Proof The proof consists in characterize how the value of functions $\mathcal{QA}(\cdot)$ and $\mathcal{E}(\cdot)$ change when two consecutive vertices are exchanged. Let us start the analysis with $\mathcal{QA}(\cdot)$ function. Consider an ordering π and two consecutive vertices u and v. Consider as well an auxiliary vertex x. First, note that if x is connected via positive edges with u and v, then the exchange between u and v does not affect the local evaluation $\mathcal{QA}(\cdot)$ into x at all, as well as if x is connected via negative edges with u and v. On the other hand, if x is connected via a positive edge with u and a negative edge with v, the exchange between u and v does affect the value of $\mathcal{QA}(\cdot)$, but the way it does it depends on whether x is arranged before or after u and v in the ordering π.

In the case when $\pi(x)<\pi(u)<\pi(v)$ and $(x,u)\in E^+\wedge(x,v)\in E^-$, an exchange between u and v increases the local evaluation $\mathcal{QA}(\cdot)$ into x by an additive factor of 2; plus one is coming from the new length of the positive edge and plus one is coming from the new length of the negative edge. The same additive factor increases this function in the case when $\pi(u)<\pi(v)<\pi(x)$ and $(x,u)\in E^-\wedge(x,v)\in E^+$. In both cases, what is happening is that the positive edge increases its length by one unit, and the negative edge decreases its length by one unit.

The opposite cases are when $\pi(x)<\pi(u)<\pi(v)$ and $(x,u)\in E^-\wedge(x,v)\in E^+$ and $\pi(u)<\pi(v)<\pi(x)$ and $(x,u)\in E^+\wedge(x,v)\in E^-$. In those cases the local evaluation $\mathcal{QA}(\cdot)$ into x decreases by an additive factor of two. What happens in those cases is that the positive edge now decreases its length by one unit, and the negative edge increases its length by one unit.

Therefore, given an ordering π, we define the following notation:

$$N^{+-}_{L(\pi)}(u,v) = \left|\left\{x : \pi(x) < \pi(u) < \pi(v) \text{ and } (x,u) \in E^+ \wedge\ (x,v) \in E^-\right\}\right|;$$
$$N^{-+}_{L(\pi)}(u,v) = \left|\left\{x : \pi(x) < \pi(u) < \pi(v) \text{ and } (x,u) \in E^- \wedge\ (x,v) \in E^+\right\}\right|;$$
$$N^{-+}_{R(\pi)}(u,v) = \left|\left\{x : \pi(u) < \pi(v) < \pi(x) \text{ and } (x,u) \in E^- \wedge\ (x,v) \in E^+\right\}\right|;$$
$$N^{+-}_{R(\pi)}(u,v) = \left|\left\{x : \pi(u) < \pi(v) < \pi(x) \text{ and } (x,u) \in E^+ \wedge\ (x,v) \in E^-\right\}\right|.$$

Moreover, for simplicity we also use this notation:

$$\#_\pi(u,v) = N^{+-}_{L(\pi)}(u,v) - N^{-+}_{L(\pi)}(u,v) + N^{-+}_{R(\pi)}(u,v) - N^{+-}_{R(\pi)}(u,v)$$

Hence, if π' is obtained from π via an exchange between u and v, we have:

$$\mathcal{QA}(\pi') = \mathcal{QA}(\pi) + 2\#_\pi(u,v). \tag{4}$$

The analysis for objective function $\mathcal{E}(\cdot)$ is similar but, the connection between the consecutive vertices u and v also plays a role. Let us see first the case when u and v are connected via a positive edge. In that case, if the auxiliary node x is of the form $\pi(x) < \pi(u) < \pi(v)$ and $(x,u) \in E^+ \wedge\ (x,v) \in E^-$ or $\pi(u) < \pi(v) < \pi(x)$ and $(x,u) \in E^- \wedge\ (x,v) \in E^+$, the number of errors produced in x increases by one. On the other hand, if x is of the form $\pi(x) < \pi(u) < \pi(v)$ and $(x,u) \in E^- \wedge (x,v) \in E^+$ or $\pi(u) < \pi(v) < \pi(x)$ or $(x,u) \in E^+ \wedge (x,v) \in E^-$ the number of errors produced in x decreases by one.

Now, when vertices u and v are connected via a negative edge, new errors or repaired errors may be in x and in u or v. Hence, now we have that, if the auxiliary node x is of the form $\pi(x) < \pi(u) < \pi(v)$ and $(x,u) \in E^+ \wedge\ (x,v) \in E^-$ or $\pi(u) < \pi(v) < \pi(x)$ and $(x,u) \in E^- \wedge (x,v) \in E^+$, the number of errors produced in x increases by one and the number of errors produced in u increases by one. On the other hand, if x is of the form $\pi(x) < \pi(u) < \pi(v)$ and $(x,u) \in E^- \wedge\ (x,v) \in E^+$ or $\pi(u) < \pi(v) < \pi(x)$ or $(x,u) \in E^+ \wedge (x,v) \in E^-$ the number of errors produced in x decreases by one and the number of errors produced in v decreases by one.

Therefore, using the same notation as before, we have:

$$\mathcal{E}(\pi') = \mathcal{E}(\pi) + \begin{cases} 2\#_\pi(u,v), & \text{if } (u,v) \in E^- \\ \#_\pi(u,v), & \text{if } (u,v) \in E^+. \end{cases} \tag{5}$$

In order to conclude the proof, assume π is a local minimum for $\mathcal{QA}(\cdot)$. Then $\#_\pi(u,v)$ is positive for all pair of consecutive vertices u, v in π. Therefore, given Eqs. (4) and (5), π is also a local minimum for $\mathcal{E}(\cdot)$. The reasoning for the opposite direction is equivalent. □

Finally, we would like to mention that, motivated in Lemma 1, we have decided to experimentally test heuristics designed for QA problem under MinSA objective function, but in cases when the input is not restricted to complete signed graphs.

4 Structural properties

In this section we introduce structural properties with respect to errors that we use later in the heuristics designed to approximate the optimal solution for MinSA problem.

We start by presenting a criterion introduced in Cygan et al. (2012) that helps to evaluate the number of errors considering precedent and subsequent vertices for a given pivot vertex.

Definition 4 Given a vertex x, a set of vertices $\mathcal{P}$ that does not contain x, and a set of vertices $\mathcal{S}$ that does not contain x, such that $\mathcal{P} \cup \{x\} \cup \mathcal{S} = V$ and $\mathcal{P} \cap \mathcal{S} = \emptyset$. We define the *weight* of the triplet $(\mathcal{P}, x, \mathcal{S})$ as follows:

$$W(\mathcal{P}, x, \mathcal{S}) := \left|\left\{(w, y) \in \mathcal{P} \times \mathcal{S} : (w, x) \in E^- \wedge (w, y) \in E^+\right\}\right| + \left|\left\{(w, y) \in \mathcal{P} \times \mathcal{S} : (x, y) \in E^- \wedge (w, y) \in E^+\right\}\right|.$$

Note that given an ordering π, if $\mathcal{P}$ and $\mathcal{S}$ are precedent and subsequent vertices of x (i.e., $\forall\, w \in \mathcal{P} \Rightarrow \pi(w) < \pi(x)$ and $\forall\, y \in \mathcal{S} \Rightarrow \pi(x) < \pi(y)$), the weight of triplet $(\mathcal{P}, x, \mathcal{S})$ does not count the number of errors in x. It counts the number of errors in which x participates but that occurs in precedent or subsequent vertices. Hence, if the first l positions of an ordering are already set (i.e., $\mathcal{P}$ is given), and position $l + 1$ will correspond to one of the vertices that remain unlabeled (i.e., x has to be chosen among $\mathcal{S}$), then a greedy criterion would prefer the x that minimizes $W(\mathcal{P}, x, \mathcal{S})$.

Note that $W(\emptyset, x, V - x) = 0$ for all x. Hence, if we follow only that criterion, any vertex can be the first vertex in the searched ordering. However, it is possible to determine whether a vertex can be the first one in an ordering without producing any error. For example, in a triangle with two positive edges and one negative edge, the vertex incident to the two positive edges can not be placed in an extreme without producing errors. While, any of the other configuration does not produce errors. This property is generalized in the following definition.

Definition 5 Given a signed graph $G = (V, E^+ \cup E^-)$, we say that a vertex $y_0 \in V$ is *k-central* if and only if there exist $2k$ vertices and an order

$$\left\{y_{-k}, y_{-(k-1)}, \ldots, y_{-1}, y_0, y_1, \ldots, y_{k-1}, y_k\right\}$$

such that this set of vertices induces a subgraph with the following edges: $(y_i, y_j) \in E^+$ if and only if $|i - j| = 1$, and $(y_i, y_j) \in E^-$ if and only if $|i - j| = k + 1$.

A k-central vertex can not be placed in the extremes. Formally stated:

Lemma 2 *Given any k-central vertex x. It holds that for any ordering of the vertices π that satisfies conditions (i) and (ii) in Theorem 1, there exist vertices w and y such that $\pi(w) < \pi(x) < \pi(y)$.*

Proof First we point out the fact that, in order to prove the lemma, it is enough to show the impossibility of the existence of an ordering π satisfying $\pi(x) < \pi(y) \forall y \in V$. In other words, in order to prove that a k-central vertex can not be placed in the extremes of an ordering, it is enough to prove that it can not be placed in the left extreme. That is because, if it is possible to place it in the right hand side extreme, then by turning the order we obtain a new order with the central vertex in the left extreme.

The proof for a general k is also by contradiction. Let us assume that there exists an ordering π that satisfies conditions (i) and (ii) such that $\pi(x) < \pi(y) \ \forall \ y \in V$, where x is a k-central vertex. Since edges (x, y_1) and $(y_{-(k-1)}, y{-}k)$ are positive, and the edge (y_1, y_{-k}) is negative, it holds that $\pi(y_1) < \pi(y_{-(k-1)})$. Otherwise, if $\pi(y_{-(k-1)}) < \pi(y_1)$, any position of vertex y_{-k} would generate a contradiction with conditions (i) or (ii). Symmetrically, it holds that $\pi(y_{-1}) < \pi(y_{(k-1)})$.

Now recursively, we show that assuming that $\pi(y_\ell) < \pi(y_{-(k-\ell)})$, it holds $\pi(y_{\ell+1}) < \pi(y_{-(k-(\ell+1))})$. Since we have proved that $\pi(y_1) < \pi(y_{-(k-1)})$, applying the recursion until $\ell = k-1$, we obtain $\pi(y_{(k-1)}) < \pi(y_{-1})$, which is a contradiction. Therefore, such an ordering does not exist.

In order to conclude the proof, let us show the recursion: given $\pi(y_\ell) < \pi(y_{-(k-\ell)})$, it holds $\pi(y_{\ell+1}) < \pi(y_{-(k-(\ell+1))})$. First, since the edge $(y_\ell, y_{(\ell+1)})$ is positive and the edge $(y_{(\ell+1)}, y_{-(k-\ell)})$ is negative, then, any ordering π that satisfies conditions (i) and (ii) must follow $\pi(y_{\ell+1}) < \pi(y_{-(k-\ell)})$. On the other hand, since the edge $(y_{(\ell+1)}, y_{-(k-\ell)})$ is negative and edge $(y_{-(k-(\ell+1))}, y_{-(k-\ell)})$ is positive, it holds that any ordering π that satisfies conditions (i) and (ii) must follow $\pi(y_{\ell+1}) < \pi(y_{-(k-(\ell+1))})$. □

Note that, an exhaustive search determines if a given vertex is 1-central in $O(n^3)$ time. In Sect. 5, we introduce two heuristics that uses an exhaustive search to decide whether a given vertex is 1-central.

5 Heuristics

In this section, we present two heuristic procedures to tackle the MinSA problem. Both heuristics can be considered as multi-start heuristics where, in each iteration, a solution is constructed and then improved with a local search method. The main difference between the two heuristics lays on the construction phase. One heuristic uses a constructive procedure based on a greedy criterion (GREEDY) and the other uses the Greedy Randomized Adaptive Search Procedure (GRASP) methodology (Feo and Resende 1989, 1995). Next, we describe both GREEDY and GRASP constructive methods and latter we detail the local search procedure.

5.1 GREEDY

The greedy constructive heuristic proposed here produces a labeling of the vertices of the graph in an ordered way. This means that at each step, GREEDY selects a vertex with a greedy criterion and assigns it to the lowest available position (e.g. 1–n). In order to determine which vertex will be the next one to be labeled, GREEDY computes the number of errors that any unlabeled vertex, not assigned yet to the solution, would produce if it is assigned to the next available position according to the Definition 4 (Sect. 4). This number of errors takes in consideration not only the assigned vertices, but also the unlabeled ones. Once it is determined for all the unlabeled vertices, GREEDY selects those which produce the minimum number or errors and adds them to a candidate list (CL). Then, GREEDY removes from CL those vertices that are 1-central with respect to the Definition 5 (Sect. 4). After that, if there are more than one vertices suitable for that position, GREEDY selects one of them randomly. Note that the latest situation can occur more than once during the construction process which could derive to different solutions when several constructions are performed.

5.2 GRASP

The second constructive heuristic is based on the ideas of GRASP methodology (Feo and Resende 1989, 1995; Resende and Ribeiro 2003). GRASP is a multi-start procedure where each iteration consists of two phases: construction and local search (Feo and Resende 1989). The GRASP procedure introduced here could be considered as an extension of the GREEDY heuristic presented above. Note that at each iteration GREEDY performs a greedy selection within the vertices in the Candidate List to select a vertex u. This vertex is then assigned to the next available label in order to produce the minimum number of errors in this iteration. Instead of that, GRASP heuristic computes a restricted candidate list (RCL) formed with good candidates selected in CL, but not only with the best ones. As it is customary in GRASP, the selection of which vertices will get into the RCL is performed according to a search parameter α which establishes a threshold. This threshold is computed as a percentage over the maximum number of possible errors in that iteration. Once the RCL is complete, GRASP selects one element in RCL at random and assigns it the lowest available label. The procedure continues until there is no unlabeled elements. The value of α is randomly selected in [0, 1] at each iteration. Note that when α equals 0, the selection is fully greedy, while when α equals 1, the selection is at random (cf. line 7 in Algorithm 1). A pseudocode of the constructive phase of GRASP is shown in Algorithm 1.

5.3 Local search

A local search procedure is able to reach a local minimum from a previously constructed solution. In this paper we propose a new local search based on *exchange moves* that will be paired later with both the GREEDY and the GRASP constructive heuristics.

Given a labeling π of the vertices of a graph and two vertices $u, v \in V$ such that $u \neq v$, an exchange move of those vertices, denoted as $\texttt{exc}(\pi, u, v)$, assigns the label of the vertex u to the vertex v in π and vice versa.

Each iteration of the local search consists of testing all possible exchanges in the solution (i.e., any vertex exchanges its position with any other vertex in the solution) trying to minimize the objective function. If there is an improvement in any exchange carried out in the iteration, the local search will perform a new whole

Algorithm 1: constructive phase of GRASP heuristic

1 **constructiveGRASP()**
2 **initialization**
3 Let L and U be the sets of labeled and unlabeled vertices of the graph;
4 Set $L = \emptyset$; $U = V$; $k = 1$;
5 **while** $U \neq \emptyset$ **do**
6 Construct a Candidate List (CL) with all vertices in U;
7 Calculate the number of errors $e(u) = W(L, u, \{U \setminus u\})\ \forall\, u \in CL$;
8 Select uniformly a random value for α in [0, 1];
9 Compute a $threshold = e_{min}(u) + \alpha(e_{max}(u) - e_{min}(u))$; /* *Where $e_{max}(u)$ and $e_{min}(u)$ are the maximum and the minimum number of errors, respectively, of any vertex $u \in CL$* */
10 Construct $RCL = \{v \in CL : e(v) \leq threshold\}$;
11 Delete from RCL those vertices that are central with respect to the vertices in U according to Definition 5; /* *Note that if the RCL becomes empty after this step, all the vertices obtained after step 9 remain in the RCL to be selected at the next step.* */
12 Select randomly a vertex u^* from RCL;
13 Label u^* with k;
14 $U = U \setminus \{u^*\}$, $L = L \cup \{u^*\}$;
15 $k = k + 1$;
16 **end while**
17 **end constructiveGRASP**

Algorithm 2: Local search

1 **localSearch**(π)
2 $improved \leftarrow$ True
3 **while** *improved* **do**
4 $improved \leftarrow$ False
5 **for** $i = 0$ ***to*** n **do**
6 **for** $j = i + 1$ ***to*** n **do**
7 $\pi' \leftarrow \mathtt{exc}(\pi, i, j)$
8 **if** $\mathcal{E}(\pi') < \mathcal{E}(\pi)$ **then**
9 $\pi = \pi'$
10 $improved \leftarrow$ True
11 **break**
12 **end if**
13 **end for**
14 **end for**
15 **end while**
16 **end localSearch**

iteration. Otherwise the local search will finish with the best solution found. The pseudocode of the local search is presented in Algorithm 2.

Let us now define a *swap* move as that one which exchange consecutive vertices in the ordering. It is easy to see that, any exchange move of two vertices $u, v \in V$ can be performed with a consecutive number of swap moves of u towards the position of v in the ordering π and vice versa. As derived from the properties presented in Sect. 4 (Proof of Lemma 2) the number of errors after a swap move is easy to update. Therefore, we performed all the exchange moves in the local search using swap moves in order to reduce the amount of time needed to reevaluate the objective function.

6 Experimental analysis

In this section we present the experiments that we carried out to test the efficiency of the heuristics presented in Sect. 5. Due to the lack of previous algorithms for the MinSA problem, we compare GRASP and GREEDY heuristics with other state-of-the-art methods for the QA problem. This is an interesting comparison due to the relation between QA and MinSA problem established in Sect. 4. In the light of preliminary experimentation, we have selected two classical heuristics from the QA problem Library (Burkard et al. 2012). Particularly, we used FANT (Taillard 1998) and GQAPSD (Feo et al. 1994) using the implementation provided by the authors. Additionally, we study the behavior of the heuristics over instances with different number of vertices, density and percentage of negative edges. All the experiments were carried out in an Intel Xeon E7420 CPU.

6.1 Instances

To test the heuristics, we used three sets of randomly generated instances named *Random*, *Interval* and *Complete*. The *Random* and *Complete* sets are formed by instances fully randomly generated (positive and negative edges). The *Interval* set is formed by graphs where the vertices and the positive edges form a unit interval graph (generated randomly) and negative edges are added on top of the unit interval positive structure (randomly generated as well). We consider such structure because the optimum value for these type of graphs is known by construction. Particularly, every instance in *Interval* set can be arranged in such a way that the number of errors is zero (Kermarrec and Thraves 2011). *Interval* and *Random* data sets are composed by 117 graphs with different number of vertices, density and percentage of negative edges. Specifically, the number of vertices ranges from 10 to 250 in steps of twenty vertices. For each number of vertices (n) we have nine graphs with different levels of edge density over the total number of edges that can appear in a graph of size n. Particularly, we consider sparse networks (with 20% of edges over the total possible), medium networks (with 50% of edges), and dense networks with (80% of edges)[2]. Once the number of edges is set, we define three different percentages of negative edges, 20, 50 and 80% of negative edges. These percentages are computed over the number of edges that appears in the graph. We have chosen these three granularities to test the influence of the number of vertices, edge density, and percentage of negative edges in the solutions found by the heuristics, as we believe that they are key factors when determining the hardness of the problem. Finally, *Complete* data set is composed by 108 *Complete* signed graphs, i.e., where every edge is present in the graph. The number of vertices ranges from 10 to 230 in steps of twenty vertices. For every amount of vertices, three different percentages of negative edges are considered, 20, 50 and 80%. We consider this last set of instances since in Sect. 3.2 it was proved an equivalence

[2] Note that these percentages are computed including positive and negative edges over the *total density* parameter.

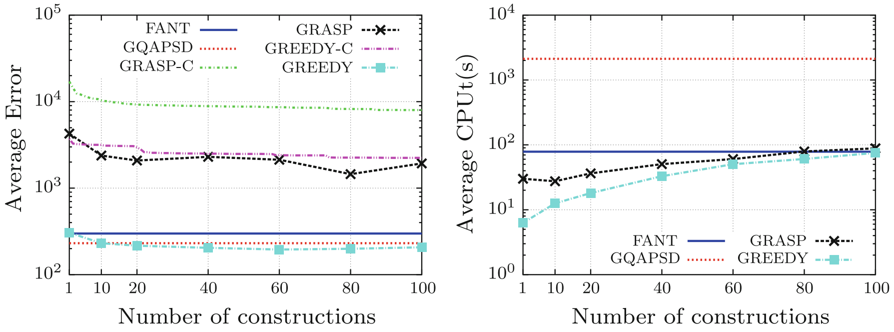

Fig. 1 Average error and average time evolution over 100 constructions for *Interval* data set

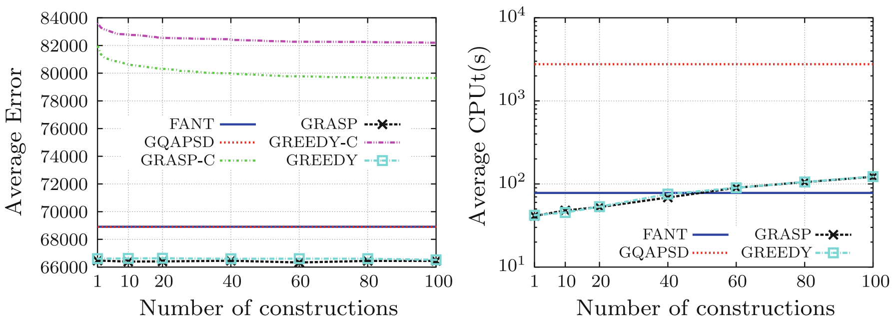

Fig. 2 Average error and average time evolution over 100 constructions for *Random* data set

between local minimums for QA and MinSA problems when the input is a complete signed graph.

6.2 Influence of the number of constructions

Note that both GREEDY and GRASP heuristics perform a constructive phase each time they are executed and then they perform a local search. The number of times the constructive phase is executed is called the *number of constructions* performed by GREEDY or GRASP heuristics. We analyze the impact of the number of constructions performed by each heuristic in both the computing time and the quality of the solution. Particularly, GREEDY and GRASP heuristics performed a determined number of constructions (1–100) and the local search was applied to the best obtained solution. Figures 1 and 2 show these results for the *Interval* and *Random* data sets, respectively. In those figures, we include: (i) two reference lines representing the quality of the solutions obtained in the constructive phase of GREEDY and GRASP (denoted in Figs. 1 and 2 by GREEDY- C and GRASP- C, respectively); (ii) two lines representing the quality of the solutions and computing time for GREEDY and GRASP heuristics; (iii) two reference lines representing the quality of the solutions and computing time for FANT and GQAPSD heuristics.

We have included lines GREEDY- C and GRASP- C in Figs. 1 and 2 in order to show the importance of the local search in the results given by those heuristics. As well, we remark that the results of the constructive phase have a monotone behavior with the number of constructions. While, the results given by GREEDY and GRASP heuristics may not follow a monotone behavior since the local search starts from a different solution in each case.

We remark that GQAPSD has slightly better results than FANT in both data sets. Nevertheless, GQAPSD computing time is at least one order of magnitude larger than FANT computing time, which makes it a not competitive alternative for bigger instances.

In the *Interval* data set, GREEDY heuristic outperformed all other strategies when 20 or more constructions were performed. Moreover, this heuristic presents the smallest average computing time. On the other hand, GRASP has a comparable average computing time but a poor quality of solutions for all number of constructions in the same data set. In both cases the average number of errors has not decreased significantly after 40 constructions.

The *Random* data set shows a different scenario. GRASP has slightly better solutions than GREEDY and similar computing time. Both heuristics present non significant improvement after 20 constructions. Additionally, both heuristics provided solutions with a less amount of errors than those provided for FANT and GAPSD heuristics. With respect to the time of computation, we note that up to 40 constructions the average computing time of the proposed heuristics is under FANT computing time (the fastest QA problem heuristic).

6.3 Comparison with the state of the art

In this subsection, we compare GREEDY and GRASP heuristics with the selected heuristics for the QA problem (FANT and GQAPSD heuristics). In Tables 1, 2 and 3 we present the results over the *Interval*, *Random* and *Complete* data sets respectively[3]. In particular, we report the average of the objective function (Avg.), the number of best values found in the experiment (#Best), the average deviation (Dev. (%)) to the best value found in the experiment and the computing time (CPUt (s)) in seconds. Additionally, for the *Interval* data set we also present the number of optimum values found (#Opt.), since it is known by construction. Notice that the objective function can be zero some times which makes impossible to calculate the deviation. To overtake this drawback we displaced the objective function by adding one unit to every solution found. This was only performed to calculate the deviation (not the average).

In the experiments reported in Tables 1, 2 and 3, GRASP and GREEDY heuristics were configured for constructing 40 solutions and improving only the best one among the constructed solutions. This configuration is derived from the previous experiment presented in Sect. 6.2 to be sure that the running time is under the running time of the state-of-the-art methods. In the set of *Interval* instances (Table 1), GREEDY heuristic

[3] A table showing all the results of the experiments can be found at http://www.cmm.uchile.cl/~mausoto/Data/signedgraphs/EmbedingSignedGraphDATA.csv.

Table 1 Comparison of the proposed methods with classical state-of-the-art methods for the QAP over the *Interval* instances

Interval (117)	GQAPSD	FANT	GREEDY	GRASP
Avg.	231	299	205	2,298
#Best	30	33	89	50
Dev. (%)	19,445.1	14,477.8	1,087.9	210,348.1
CPUt (s)	2,120	78	33	50
#Opt.	18	17	82	45

Table 2 Comparison of the proposed methods with classical state-of-the-art methods for the QAP over the *Random* instances

Random (117)	GQAPSD	FANT	GREEDY	GRASP
Avg.	68,902.6	68,913.9	66,460.71	66,600.6
Dev. (%)	26.1	26.2	3.3	1.3
#Best	6	6	57	68
CPUt (s)	2,761	78	69	76

Table 3 Comparison of the proposed methods with classical state-of-the-art methods for the QAP over the *Complete* instances

Complete (108)	GQAPSD	FANT	GREEDY	GRASP
Avg.	178,430.6	177,718.3	177,481.5	177,166.5
Dev. (%)	1.2	1.1	1.2	0.7
#Best	17	22	46	51
CPUt (s)	4,222.9	29.1	78.2	75.9

provides the higher number of best values found in the experiment (89 out of 117 instances) and has the smaller deviation among the compared heuristics. Additionally, it is the fastest method. It is also important to remember that for this set of instances the optimum value is known by construction. Specifically, GREEDY found 82 optima values. Nevertheless, in the *Random* set of instances GRASP heuristic has a better deviation with respect to the best known value (see Table 2), higher number of best values found and a smaller average value of the objective function. Our arguments for the occurrence of this phenomenon are based in the fact that *Interval* instances have an optimum arrangement with zero errors. Hence, at every step of the construction there exists at least one vertex whose weight function is zero. Therefore, while GREEDY heuristic chooses that vertex at each step of the construction phase, GRASP heuristic may not do it. Then, if at the very first steps of the construction GREEDY made the good choices, it will be directed to the optimum solution. On the other hand, at every step of the GRASP construction, it may take a bad decision that will bring it to a really bad solution. The previous fact is not necessarily true in the *Random* set of instances where at each step the best choice may not be the one with minimum weight value at that step. Therefore, GREEDY heuristic is induced to produce more errors in this case. Finally, in the *Complete* data set, again GRASP obtained the higher number of best values and the smallest deviation among the tested methods. It is important to remark that over this data set all the methods performed similarly. However, statistical analysis carried out below shows differences among the heuristics.

To complement this information, we have applied the Friedman test to the best solutions obtained by each of the four heuristics. This experiment was performed separately for the *Interval*, *Random* and *Complete* data sets. In all the cases the resulting p value of 0.000 obtained clearly indicates that there are statistically significant differences among the four methods tested. A typical post-test analysis consists of ranking the methods under comparison according to the average rank values computed with this test. According to this, the best method in the *Interval* data set is GREEDY (with a rank value of 1.88), followed by the GRASP (2.66), GQAPSD (2.68) and finally FANT (with 2.78 rank value). With respect to the *Random* data set, GRASP was the best method (rank value 1.54) followed by GREEDY (1.64), FANT (3.35) and GQAPSD (3.47). Finally, in the *Complete* data set, GRASP was again the best method (rank value 2.08) followed by GREEDY (2.43), FANT (2.64) and GQAPSD (2.85). Additionally, we compared the two best procedures (GRASP and GREEDY) with the Wilcoxon test which measures whether the solutions come or not from different methods (i.e., they are significantly different). Considering each data set separately, the obtained p value of 0.000 over the *Interval* data set confirms that there are differences between both methods resulting GREEDY as the best one. On the other hand, the p value of 0.187 obtained over the *Random* and the p value of 0.081 obtained over the *Complete* data set indicates that there are not statistical significative differences between both methods for these data sets. Consider all the data sets together, we can affirm that there is not statistical significative differences between GREEDY and GRASP.

6.4 Behavior of the heuristics with respect to the edge density

In this section, we study the impact of different graph properties in the quality of the solutions given by the proposed heuristics. We recall that *Interval* and *Random* sets of instances contain graphs with different edge density: sparse (20%), medium (50%) and dense (80%). For each of these densities, we construct graphs with a varied proportion of negative edges: 20, 50 and 80%.

Figure 3 shows the average values for the objective function in the *Interval* set with respect to density (left) and negative edge proportion (right). We note that for sparse graphs, all heuristics has a good performance, additionally QA problem heuristics have slightly better results. Remarkably, QA problem and GRASP heuristics provide worse results in medium dense graphs. The only exception is GREEDY, which found the optimum value for almost all medium and dense graphs. On the other hand, proportion of negative edges (Fig. 3 right) has a different influence on the objective value. While for QA problem heuristics this proportion increases the average number of errors, for GREEDY heuristic the behavior is stable. Particularly, GREEDY heuristic provides better solutions when the proportion of negative edges is larger.

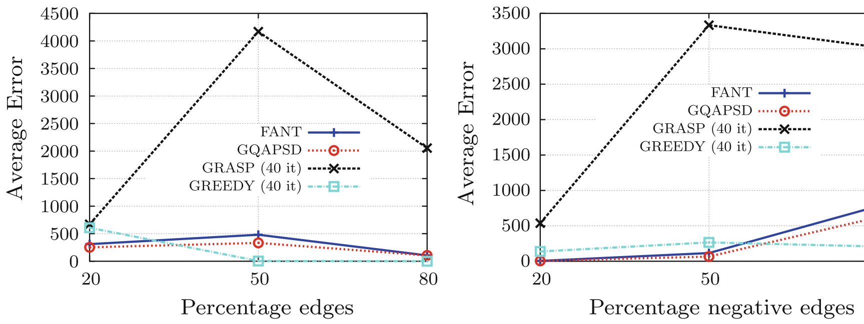

Fig. 3 Average error comparison for heuristics (40 iterations) in function of the edge density and negative edge proportion (*Interval* data set)

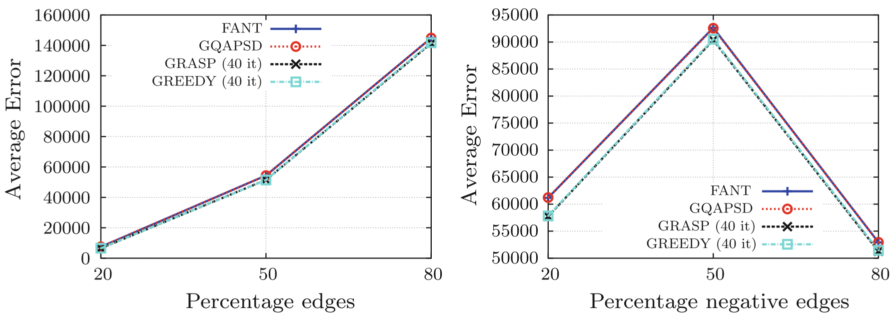

Fig. 4 Average error comparison for heuristics (40 iterations) in function of the edge density and negative edge proportion (*Random* data set)

In the *Random* set (Fig. 4) all heuristics have a similar behavior with no clear differences. The objective function grows with edge density since it reaches its maximum value for the set of instances with half of negatives edges and the minimum value for graphs with a 80% of edges with negative value. In the *Complete* set, the behavior of all heuristics is similar to the one showed in *Random* set. For the sake of simplicity we do not include lines depicting such behavior.

7 Conclusions and future work

In this work, we have studied MinSA problem in the line. We proved that when the input graph is complete, local minimums for MinSA problem are equivalent to local minimums for a special case of the well known QA problem. We aimed to approximate via heuristics the optimal solution for MinSA problem. Particularly, we proposed two heuristics based on a greedy criterion, where the construction phase of one of them is according to GRASP methodology. We experimentally compared the proposed heuristics together with two heuristics designed for QA problem. In the experiments, we have seen that the GREEDY heuristic provides the optimum value most of the times when the input graph has an optimal solution with zero errors. Such a conclusion is valuable since it tells us that the GREEDY heuristic works well at the moment of recognition of graphs that have an optimal solution with zero errors, problem that is NP-complete (Cygan et al. 2012). On the other hand, GRASP heuristic provides slightly better results when the input graph has no special structure. Even though, in this case, statistical analysis tell us that there is no significant differences between GRASP and GREEDY. Also, the heuristics designed for QA problem provide comparable results under MinSA objective function, allowing us to claim a strong connection between MinSA and its quadratic formulation.

Concerning the future work, we would like to explore the following questions: (a) Are global minimums for MinSA problem equivalent to global minimums of its quadratic formulation when the input graph is complete? (b) Are global minimums for the QA version global minimums for MinSA problem for general input graphs? (we remark that, in the second question, the converse does not apply). We also consider appealing understand the complexity of MinSA problem with respect to density of edges because it can be a way to parametrize the complexity of the problem.

In addition, due to the effectiveness of the GREEDY heuristic in the recognition of graphs with an optimal solution with zero errors, it would be interesting to investigate the impact of discard k-central vertices for bigger values of k (recall that, in this work, the GREEDY heuristic discards only 1-central vertices as extreme vertices). Though, performing a deeper discarding of k-central vertices shall impact in the complexity of the algorithm. Therefore, it might be interesting to search for the good trade-off between recognition capabilities versus complexity.

Acknowledgments Christopher Thraves is supported by Spanish MICINN Grant Juan de la Cierva, Comunidad de Madrid Grant S2009TIC-1692 and Spanish MICINN Grant TIN2008–06735-C02-01. Eduardo G. Pardo is supported by Spanish MICINN Grant TIN2008-06890-C02-02 and TIN2012-35632-C02-02.

References

Antal T, Krapivsky PL, Redner S (2005) Dynamics of social balance on networks. Phys Rev E 72(3):036–121

Bansal N, Blum A, Chawla S (2004) Correlation clustering. Mach Learn 56(1–3):89–113

Burkard R, Çela E, Karisch S, Rendl F (2012) Qaplib: a qudratic assignment problem library. http://www.opt.math.tu-graz.ac.at/qaplib/

Burkard R, Çela E, Pardalos P, Pitsoulis L (1998) The quadratic assignment problem. Kluwer Academic Publishers, Dordrecht

Burkard R, Rendl F (1984) A thermodynamically motivated simulation procedure for combinatorial optimization problems. Eur J Oper Res 17(2):169–174

Cartwright D, Harary F (1956) Structural balance: a generalization of Heider's theory. Psychol Rev 63(5):277–293

Çela E (1998) The quadratic assignment problem: theory and algorithms. Kluwer Academic Publishers, Dordrecht

Commander CW (2007) A survey for the quadratic assignment problem. Eur J Oper Res 176(2):657–690

Connolly DT (1990) An improved annealing scheme for the qap. Eur J Oper Res 46(1):93–100

Cygan M, Pilipczuk M, Pilipczuk M, Wojtaszczyk JO (2012) Sitting closer to friends than enemies, revisited. In: Proceedings of the 37th international symposium on mathematical foundations of computer science (MFCS) (2012)

Davis JA (1967) Clustering and structural balance in graphs. Hum Relat 20(2):181–187

Drezner Z, Hahn PM, Taillard E (2005) Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. Ann Oper Res 139(1):65–94

Feo T, Resende M (1989) A probabilistic heuristic for a computationally difficult set covering problem. Oper Res Lett 8:67–71

Feo T, Resende M (1995) Greedy randomized adaptive search procedures. J Glob Optim 6:109–133
Feo T, Resende M, Smith S (1994) A greedy randomized adaptive search procedure for maximum independent set. Oper Res 42:860–878
Harary F (1953) On the notion of balance of a signed graph. Mich Math J 2(2):143–146
Harary F, Kabell JA (1980) A simple algorithm to detect balance in signed graphs. Math Soc Sci 1(1): 131–136
Kermarrec AM, Thraves C (2011) Can everybody sit closer to their friends than their enemies? In: Proceedings of the 36th international symposium on mathematical foundations of computer science (MFCS), pp 388–399
Koopmans TC, Beckmann M (1957) Assignment problems and the location of economic activities. Econometrica 25(1):53–76
Kunegis J, Schmidt S, Lommatzsch A, Lerner J, Luca EWD, Albayrak S (2010) Spectral analysis of signed graphs for clustering, prediction and visualization. In: Proceedings of the SIAM international conference on data mining (SDM), pp 559–571
Leskovec J, Huttenlocher DP, Kleinberg J (2010) Predicting positive and negative links in online social networks. In: Proceedings of the 19th international conference on world wide web (WWW), pp 641–650
Leskovec J, Huttenlocher DP, Kleinberg J (2010) Signed networks in social media. In: Proceedings of the 28th international conference on human factors in computing systems (CHI), pp 1361–1370
Loiola EM, de Abreu NMM, Boaventura-Netto PO, Hahn P, Querido T (2007) A survey for the quadratic assignment problem. Eur J Oper Res 176(2):657–690
Nehi HM, Gelareh S (2007) A survey of meta-heuristic solution methods for the quadratic assignment problem. Appl Math Sci 1(45–48):2293–2312
Pardalos PM, Rendl F. Wolkowicz H (1994) The quadratic assignment problem: a survey and recent developments. In: Proceedings of the DIMACS workshop on quadratic assignment problems, volume 16 of DIMACS series in discrete mathematics and theoretical computer science. American Mathematical Society, Providence, pp 1–42
Resende M, Ribeiro C (2003) Greedy randomized adaptive search procedures. In: Glover F, Kochenberger FS, Hillier CC, Price (eds) Handbook of metaheuristics, international series in operations research and management science, vol 57. Springer, New York, pp 219–249
Sahni S, Gonzalez T (1976) P-complete approximation problems. J ACM 23(3):555–565
Skorin-Kapov J (1990) Tabu search applied to the quadratic assignment problem. ORSA J Comput 2(1): 33–45
Szell M, Lambiotte R, Thurner S (2010) Multirelational organization of large-scale social networks in an online world. Proc Natl Acad Sci USA (PNAS) 107(31), 13, 636–13, 641 (2010)
Taillard E (1991) Robust taboo search for the quadratic assignment problem. Parallel Comput 17(45): 443–455
Taillard E (1998) Fant: fast ant system. Technical report, Dalle Molle Institute for Artificial Intelligence, Lugano