Between a rock and a hard place: the two-to-one assignment problem

Dries Goossens, Sergey Polyakovskiy, Frits C. R. Spieksma & Gerhard J. Woeginger

Mathematical Methods of Operations Research

ISSN 1432-2994
Volume 76
Number 2

Math Meth Oper Res (2012) 76:223-237
DOI 10.1007/s00186-012-0397-2

Springer

Math Meth Oper Res (2012) 76:223–237
DOI 10.1007/s00186-012-0397-2

ORIGINAL PAPER

Between a rock and a hard place: the two-to-one assignment problem

Dries Goossens · Sergey Polyakovskiy · Frits C. R. Spieksma · Gerhard J. Woeginger

Received: 30 November 2010 / Accepted: 27 June 2012 / Published online: 1 August 2012

Abstract We describe the two-to-one assignment problem, a problem in between the axial three-index assignment problem and the three-dimensional matching problem, having applications in various domains. For the (relevant) case of decomposable costs satisfying the triangle inequality we provide, on the positive side, two constant factor approximation algorithms. These algorithms involve solving minimum weight matching problems and transportation problems, leading to a 2-approximation, and a $\frac{3}{2}$-approximation. Moreover, we further show that the best of these two solutions is a $\frac{4}{3}$-approximation for our problem. On the negative side, we show that the existence of a polynomial time approximation scheme for our problem would imply P = NP. Finally, we report on some computational experiments showing the performance of the described heuristics.

This research has been supported by the Netherlands Organisation for Scientific Research (NWO) under grant 639.033.403, by BSIK grant 03018 (BRICKS: Basic Research in Informatics for Creating the Knowledge Society), and by OT Grant OT/07/015. An earlier version of this work appeared in the Proceedings of the 7th Workshop on Approximation and Online Algorithms (WAOA 2009).

D. Goossens · S. Polyakovskiy · F. C. R. Spieksma (✉)
Operations Research Group, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium
e-mail: frits.spieksma@kuleuven.be

D. Goossens
e-mail: dries.goossens@kuleuven.be

S. Polyakovskiy
e-mail: sergey.polyakovskiy@kuleuven.be

G. J. Woeginger
Department of Mathematics, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, The Netherlands
e-mail: gwoegi@win.tue.nl

Keywords Assignment problem · Matching problem · Efficient algorithm · Approximation

1 Introduction

The two-to-one assignment problem 2-1-AP in the title of this paper is defined as follows: Given are a set R of $2n$ red elements and a set G of n green elements. A *feasible triple* (or just *triple*, for short) consists of two distinct elements from R and of a single element from G. There is a cost-coefficient c_{ijk} given for each feasible triple, where the indices i and j run over the set R, and the index k runs over the set G. The goal is to select n triples such that each element from the ground set $R \cup G$ is used exactly once, and such that the sum of all triple costs is minimized.

The two-to-one assignment problem is closely related to the three-dimensional matching problem (3DM) and to the axial three-index assignment problem (3AP), which play the role of rock and hard place in the title. In the 3DM a single set of $3n$ elements is given, and any three elements constitute a feasible triple. In the 3AP three n-element sets are given, and a feasible triple consists of one element from each of the three sets. We observe that problem 2-1-AP (i) is a special case of 3DM, and (ii) does contain 3AP as a special case. To see (i), observe that adding the 'missing' cost-coefficients with a sufficiently high cost to an instance of 2-1-AP produces an equivalent instance of 3DM. An analogous observation turns any instance of 3AP into an equivalent instance of 2-1-AP, and thus yields (ii). Hence, with respect to the operation of adding missing cost-coefficients, the two-to-one assignment problem 2-1-AP is sandwiched between the rock 3AP and the hard place 3DM.

From the practical point of view, any setting where one selects triples that consist of two elements from one set and a single element from another set can be modeled as problem 2-1-AP. We list several applications from diverse areas:

Satellite refueling. Servicing and refueling spacecraft in orbit extends the lifetime of the spacecraft, reduces launching and insurance cost, and increases operational flexibility and robustness. Dutta and Tsiotras (2008) investigate a scenario where satellites exchange fuel amongst themselves in pairs. This amounts to pairing up $2n$ satellites and to assigning these pairs to n locations in orbit.

Chromosome pairing. A human cell contains two homologous chromosomes from each of the 22 chromosome classes known as the autosomes, and two sex chromosomes from the X and the Y class depending on the gender of the individual. Biyani et al. (2005) investigate a joint classification and pairing problem, where $2n$ chromosomes have to be paired in homologues and then assigned to n autosome classes. The cost-coefficients rely on statistical properties of chromosome data and encode certain maximum likelihood estimates.

Sports scheduling. Urban and Russel (2003) discuss the scheduling of football competitions that take place on n venues that are not associated with any of the $2n$ participating teams. Apart from financial constraints, the cost-coefficients of the triples also encode travel distances and fan-related issues.

Gender matching. Back in 1963, Brian Wilson wrote the song "*Surf City*", which deals with a community of surfers in the California of the 1960s. The song text contains the well-known line "*Two girls for every boy!*". A recent cover version by the Go-Gos changed the lyrics into "*Two boys for every girl!*". Problem 2-1-AP covers a variety of scenarios in similar non-monogamous societies.

The cost-coefficients in the above applications are not arbitrary numbers; instead these cost-coefficients display some structure. Here, we will look into the special case DECOM of 2-1-AP, where there is a non-negative symmetric distance d_{ij} specified for every pair (i, j) of elements in $R \cup G$. The cost-coefficients of feasible triples (i, j, k) are *decomposable*, which means that they are defined as

$$c_{ijk} = d_{ij} + d_{ik} + d_{jk}. \tag{1}$$

If additionally these distances d_{ij} satisfy the triangle inequality

$$d_{ij} \le d_{ik} + d_{jk} \qquad \text{for all } i, j, k \tag{2}$$

then we denote the resulting special case of 2-1-AP as problem Δ-DECOM.

Let us first recall several special cases of 3DM and 3AP from the literature, where the underlying cost-coefficients are decomposable in a similar fashion; notice however, that in the case of cost-coefficients satisfying (1) and (2), the reductions from 3AP to 2-1-AP, and from 2-1-AP to 3DM sketched above no longer work. Crama and Spieksma (1992) showed that 3AP with decomposable costs (1) does not admit any polynomial time constant factor approximation algorithm, unless P = NP. However if the distances also satisfy the triangle inequality (2), then a polynomial time 4/3-approximation algorithm becomes possible. Spieksma and Woeginger (1996) prove that 3AP with decomposable costs remains NP-hard when the distances d_{ij} are the Euclidean distances of a set of points in the Euclidean plane. Magyar et al. (2000) describe genetic algorithms for 3DM with decomposable costs of the form $c_{ijk} = \min\{d_{ij} + d_{ik}, d_{ij} + d_{jk}, d_{ik} + d_{jk}\}$. Burkard et al. (1996) deal with a variant of 3AP with decomposable costs of the form $c_{ijk} = a_i b_j c_k$; this special case is NP-hard and does not admit any polynomial time constant factor approximation algorithm unless P = NP. Finally, Goossens et al. (2010) give a 2-approximation algorithm for a bottleneck version of 3AP with decomposable costs, and this is shown to be best-possible (unless P = NP).

1.1 Our results

The results of Crama and Spieksma (1992) for 3AP easily imply that DECOM is NP-hard, and that DECOM does not admit any polynomial time constant factor approximation algorithm unless P = NP. Therefore, we will mainly concentrate on the approximation of the more tractable variant Δ-DECOM. We will derive the following results.

- Problem Δ-DECOM is APX-hard, and thus cannot possess a PTAS unless P = NP (Sect. 2).
- We exhibit a polynomial time 4/3-approximation algorithm for Δ-DECOM (Sect. 3). This extends and generalizes the results of Crama and Spieksma (1992).

Table 1 Current approximability status of 3DM, 2-1-AP, and 3AP

	3DM		2-1-AP		3AP	
	+	−	+	−	+	−
Decomposable		No cf (ff Crama and Spieksma 1992)		No cf (ff Crama and Spieksma 1992)		No cf (Crama and Spieksma 1992)
Δ-inequality		APX (ff Sect. 2)	$\frac{4}{3}$ (Sect. 3)	APX (Sect. 2)	$\frac{4}{3}$ (Crama and Spieksma 1992)	APX (ff Sect. 2)

Further, in Sect. 4, we give some computational evidence of the performance of the approximation algorithms by running them on randomly generated instances.

We summarize the current state of affairs in Table 1, where the row 'decomposable' refers to the relevant problem with decomposable cost-coefficients, and the row 'Δ-inequality' refers to the special case of distances satisfying the triangle inequality. A column marked with '+' contains the worst-case ratio's achieved, a column marked '−' describes what cannot be attained in polynomial time, unless P = NP. In particular, 'no cf' stands for 'no constant factor approximation algorithm, unless P = NP, 'APX' stands for 'APX-hard', and 'ff' stands for 'follows from'.

2 APX-hardness of Δ-DECOM

In this section we prove that Δ-DECOM is APX-hard. Our proof is through an approximation preserving reduction from the following problem.

Problem: Maximum bounded 3-dimensional matching (MAX- 3DM- B)

Instance: Three sets $X = \{x_1, \ldots, x_q\}$, $Y = \{y_1, \ldots, y_q\}$, and $Z = \{z_1, \ldots, z_q\}$. A subset $T \subseteq X \times Y \times Z$ such that any element in X, Y, Z occurs in one, two or three triples in T; note that this implies $q \leq |T| \leq 3q$.

Goal: Find a subset T' of T of maximum cardinality such that no two triples of T' agree in any coordinate; such sets of triples are called *matchings*.

Kann (1991) established that MAX- 3DM- B is APX-hard. An instance of MAX- 3DM- B is called a *perfect* instance, if its optimal solution consists of q triples that cover all elements in $X \cup Y \cup Z$. Petrank (1994) proved that even perfect instances of MAX- 3DM- B are hard to approximate, and that the existence of a PTAS for perfect instances would imply P = NP.

The rest of this section is dedicated to the APX-hardness argument for Δ-DECOM. Starting from an arbitrary instance I' of MAX- 3DM- B, we will build a corresponding instance I of Δ-DECOM by using the gadget depicted in Fig. 1. We note that this gadget is a simplification of another gadget that has been designed by Garey and Johnson (1979) to establish NP-hardness of the problem PARTITION INTO TRIANGLES.

- For each element of $X \cup Y \cup Z$ in instance I' of MAX- 3DM- B, there is a corresponding point in instance I of Δ-DECOM; these points are called *element points*.

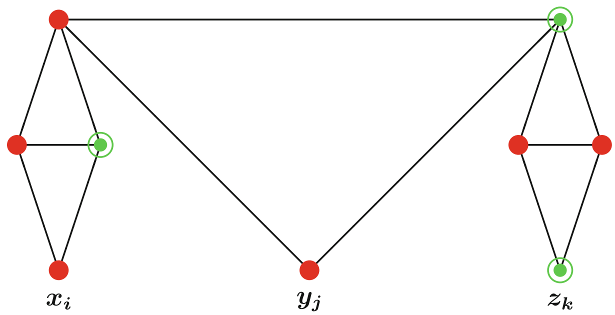

Fig. 1 The gadget in the APX-hardness proof of Δ-DECOM

- For each triple (x_i, y_j, z_k) in T, there are six corresponding points in instance I. These six points are called *gadget points*, and they are connected through auxiliary edges to the three element points corresponding to x_i, y_j, z_k as indicated in Fig. 1.

The sets R and G in instance I are defined according to the colors in Fig. 1; in particular element points that correspond to elements of $X \cup Y$ are red, and element points that correspond to elements of Z are green. What about the distances $d(\cdot, \cdot)$? If two points are connected by an auxiliary edge in the gadget in Fig. 1, then they are at distance 1; otherwise their distance equals 2. Note that these distances satisfy the triangle inequality, and note that any two gadget points from different gadgets are at distance 2. This completes the description of the instance I of Δ-DECOM.

Note that instance I contains $3q + 6|T|$ points. Any feasible solution partitions the points into $q + 2|T|$ triangles of perimeters 3, 4, 5, and 6. Triangles of perimeter 3 are called *good*, and triangles of perimeter 4 or more are called *bad*. The following observation will be useful.

Observation 2.1 *Consider a feasible solution, in which all nine points in Fig. 1 belong to good triangles. Then either each of the three element points is matched with two gadget points from this gadget, or none of them is matched with a point from this gadget.*

The following two lemmas are easy consequences of Observation 2.1.

Lemma 2.2 *Instance I' of* MAX- 3DM- B *possesses a matching of size q, if and only if the constructed instance I of Δ-DECOM satisfies* $\mathrm{OPT}(I) = 3q + 6|T|$.

Lemma 2.3 *Let $\delta \geq 0$ be a real number. If instance I has a feasible solution of cost at most $3q + 6|T| + \delta q$, then instance I' possesses a matching of size at least $(1 - 4\delta)q$.*

Proof Consider some feasible solution for instance I. We call a gadget *damaged*, if at least one of its six gadget points lies in a bad triangle. We call an element point *damaged*, if (i) it is in a bad triangle [type (i)], or if (ii) it is in a good triangle but together with a gadget point from a damaged gadget [type (ii)].

Fix now a feasible solution for instance I with cost at most $3q + 6|T| + \delta q$. Clearly, since a bad triangle has perimeter at least 4, and a good triangle has perimeter at most 3, it follows that at most δq of the triangles in this feasible solution are bad.

Thus, there are at most $3\delta q$ damaged element points of type (i). Furthermore, the fact that there are at most δq bad triangles also implies that there are at most $3\delta q$ damaged gadgets, each of which may yield at most three damaged element points of type (ii). Hence, altogether there are at most $12\delta q$ damaged element points, which leads to at least $3(1-4\delta)q$ undamaged element points. Every undamaged element point is in a triangle with two gadget points from the same undamaged gadget, and—since the gadget is undamaged—there are two other undamaged element points that are in triangles with points from the very same gadget. Hence the $3(1-4\delta)q$ undamaged element points can be divided into groups of three that correspond to $(1-4\delta)q$ undamaged gadgets. Then the corresponding $(1-4\delta)q$ triples in instance I' form a matching. □

In case we start the construction from a perfect instance I' of MAX- 3DM- B, Lemma 2.2 yields $\text{OPT}(I) = 3q + 6|T|$ for the resulting instance I. A $(1+\varepsilon)$-approximation algorithm for Δ-DECOM would yield an approximate objective value of at most

$$(1+\varepsilon)\cdot \text{OPT}(I) \;\leq\; 3q + 6|T| + 21\varepsilon \cdot q.$$

Here we have applied $|T| \leq 3q$. Then Lemma 2.3 (with $\delta = 21\epsilon$) yields a matching of size at least $(1-84\epsilon)q$ for instance I'. Hence, a PTAS for Δ-DECOM would imply a PTAS for perfect instance of MAX- 3DM- B.

Theorem 2.4 *Δ-DECOM is APX-hard.*

Similar arguments can be used to show that also the special cases of problems 3AP and 3DM with decomposable costs with $d_{ij} \in \{1, 2\}$ are APX-hard.

3 Approximation results for Δ-DECOM

We formulate and analyze three polynomial time approximation algorithms for Δ-DECOM: We first design a 2-approximation algorithm, then a 3/2-approximation algorithm, and finally combine these two algorithms to get a 4/3-approximation. Our approach adds a number of new ideas to the work of Crama and Spieksma (1992). Our methods involve solving assignment problems, weighted matching problems, and transportation problems; we refer to Schrijver (2003) for an overview of methods and achievable time complexities for these problems.

3.1 The transportation heuristic

Consider some instance I of Δ-DECOM. Our first heuristic TP uses the following transportation problem as a main ingredient:

$$\min \sum_{i\in R}\sum_{k\in G} d_{ik}x_{ik}$$

$$\text{s.t.} \quad \sum_{i\in R} x_{ik} = 2 \quad \text{for all } k \in G$$

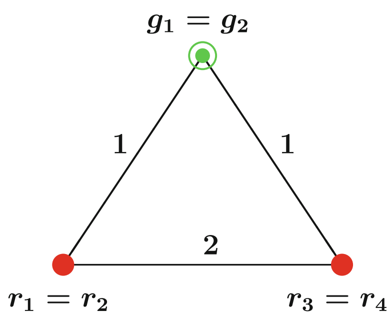

Fig. 2 A worst case instance for heuristic TP

$$\sum_{k \in G} x_{ik} = 1 \quad \text{for all } i \in R$$

$$x_{ik} \in \{0, 1\} \quad \text{for all } i \in R, \text{for all } k \in G.$$

Let x^* denote an optimal solution to this transportation problem, which assigns to every element of G two distinct elements from R. Then heuristic TP returns the corresponding feasible solution $X = \{(i, j, k) : x^*_{ik} = 1, x^*_{jk} = 1\}$. For the analysis of TP we fix an optimal set Z of feasible triples for instance I. We deduce successively:

$$\text{TP}(I) = \sum_{(i,j,k) \in X} (d_{ij} + d_{ik} + d_{jk}) \;\le\; 2 \sum_{(i,j,k) \in X} (d_{ik} + d_{jk}) \tag{3}$$

$$\le 2 \sum_{(i,j,k) \in Z} (d_{ik} + d_{jk}) \;\le\; 2 \cdot \text{OPT}(I). \tag{4}$$

Here the inequality in (3) follows by applying the triangle inequality in order to bound d_{ij}. The first inequality in (4) holds since the transportation problem minimizes the underlying value $\sum(d_{ik} + d_{jk})$. The second inequality in (4) is trivial.

To see that equality may be attained in (4), consider the instance I depicted in Fig. 2. This instance has $R = \{r_1, r_2, r_3, r_4\}$ and $G = \{g_1, g_2\}$ with distances as indicated in the figure. The optimal solution $Z = \{(r_1, r_2, g_1), (r_3, r_4, g_2)\}$ has cost $\text{OPT}(I) = 4$. If the transportation problem assigns r_1 and r_3 both to g_1 and assigns r_2 and r_4 both to g_2, then TP ends up with $X = \{(r_1, r_3, g_1), (r_2, r_4, g_2)\}$, and hence $\text{TP}(I) = 8$. All in all, this yields the following theorem.

Theorem 3.1 *Heuristic TP is a polynomial time* 2*-approximation algorithm for problem* Δ*-DECOM. Moreover, there exist instances* I *for which* $TP(I) = 2\,\text{OPT}(I)$.

3.2 The match-and-assign heuristic

The match-and-assign heuristic MA for Δ-DECOM goes through two stages that both solve a minimum weight matching problem: In the first stage, MA computes a minimum weight matching M for the $2n$ elements in R under the distances d_{ij} with $i, j \in R$. In the second stage, MA computes a minimum weight assignment of the pairs (i, j) in M to the elements k in G under the costs c_{ijk}. This second stage can be described by the following integer program:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in M} \sum_{k\in G} c_{ijk} x_{ijk} \\
\text{s.t.} \quad & \sum_{(i,j)\in M} x_{ijk} = 1 && \text{for all } k \in G \\
& \sum_{k\in G} x_{ijk} = 1 && \text{for all } (i, j) \in M \\
& x_{ijk} \in \{0, 1\} && \text{for all } (i, j) \in M, k \in G.
\end{aligned}$$

For an optimal solution x^* of this assignment problem, heuristic MA returns the feasible solution $X = \{(i, j, k) : x^*_{ijk} = 1\}$. For the analysis of MA we fix an optimal solution Z of I.

Lemma 3.2 *There exists a partition of R into two subsets R_1 and R_2 with $|R_1| = |R_2| = n$ that has the following properties.*

- *Every pair $(i, j) \in M$ has one element in R_1, and the other element in R_2.*
- *In the optimal solution Z, every element $k \in G$ is in a triple with one element called $Z_1(k)$, in R_1 and one element called $Z_2(k)$, in R_2.*

Proof We construct a multi-graph with $2n$ edges on the vertex set R (recall that a multi-graph is graph that may contain parallel edges). The edge set contains all n edges in M; these edges are called matching-edges. Furthermore, for every triple (i, j, k) in the optimal solution Z there is a corresponding edge between i and j; these edges are called triple-edges. Since every vertex in this multi-graph is incident to exactly one matching-edge and one triple-edge, the multi-graph is a collection of even cycles.

We orient the edges along every cycle, so that every vertex has precisely one incoming and one out-going arc (the resulting orientation is not unique). If a triple-edge corresponding to triple (i, j, k) is oriented from i to j, then we define $Z_1(k) = i$ and $Z_2(k) = j$, and we put i into R_1 and j into R_2. This construction satisfies all desired properties. □

We define another feasible solution Y that consists of n triples from $R_1 \times R_2 \times G$: A triple (i, j, k) is in Y, if and only if $(i, j) \in M$ and $Z_2(k) = j$. This yields

$$\text{MA}(I) = \sum_{(i,j,k)\in X} c_{ijk} \le \sum_{(i,j,k)\in Y} c_{ijk} \tag{5}$$

$$= \sum_{(i,j,k)\in Y} (d_{ij} + d_{ik} + d_{jk}) \le 2 \sum_{(i,j,k)\in Y} (d_{ij} + d_{jk}) \tag{6}$$

$$= 2 \sum_{(i,j,k)\in Y} d_{ij} + 2 \sum_{(i,j,k)\in Z} d_{jk} \le 2 \sum_{(i,j,k)\in Z} (d_{ij} + d_{jk}). \tag{7}$$

Here the inequality in (5) follows, since the second stage of MA matches the pairs in M at minimum cost with the elements of G, whereas Y matches them according

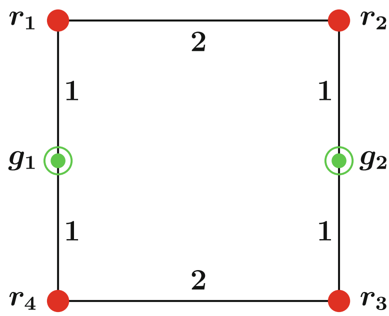

Fig. 3 A worst case instance for heuristic MA

to $Z_2(\cdot)$. The inequality in (6) follows by applying the triangle inequality in order to bound d_{ik}. Finally the inequality in (7) follows from the fact that matching M is the minimum cost matching for the set R.

Next, by a similar argument we get the following inequality that is perfectly symmetric to inequality (7):

$$\mathrm{MA}(I) \leq 2 \sum_{(i,j,k)\in Z} (d_{ij} + d_{ik}). \tag{8}$$

Adding (7) to (8) and using the triangle inequality entails

$$\begin{aligned} \mathrm{MA}(I) &\leq \sum_{(i,j,k)\in Z} (2d_{ij} + d_{ik} + d_{jk}) \\ &\leq \sum_{(i,j,k)\in Z} \frac{3}{2}(d_{ij} + d_{ik} + d_{jk}) = \frac{3}{2} \cdot \mathrm{OPT}(I). \end{aligned} \tag{9}$$

To see that equality may hold in (9), consider the instance I in Fig. 3. This instance has $R = \{r_1, r_2, r_3, r_4\}$ and $G = \{g_1, g_2\}$. The distances are as indicated in the picture; whenever two elements are not connected by an edge, their distance equals 2. An optimal solution is $Z = \{(r_1, r_4, g_1), (r_2, r_3, g_2)\}$ with cost $\mathrm{OPT}(I) = 8$. In the first stage, heuristic MA may find the matching $M = \{(r_1, r_2), (r_3, r_4)\}$. Then the second stage yields $X = \{(r_1, r_2, g_1), (r_3, r_4, g_2)\}$, and $\mathrm{MA}(I) = 12$. We summarize our results in the following theorem.

Theorem 3.3 *Heuristic MA is a polynomial time* $3/2$*-approximation algorithm for problem* Δ*-DECOM. Moreover, there exist instances I for which* $MA(I) = (3/2)\,\mathrm{OPT}(I)$.

3.3 The combined heuristic

Do there exist instances of Δ-DECOM on which both TP and MA perform poorly (that is, close to their worst case performance guarantees)? We will demonstrate that

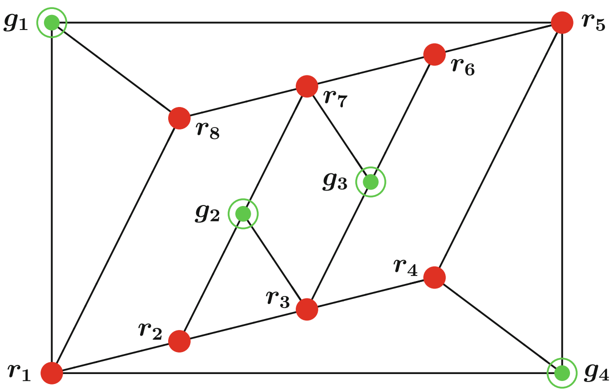

Fig. 4 A worst case instance for heuristic COMB

the answer is actually no. The combined heuristic COMB runs TP and MA on instance I, and then outputs the best of the two solutions.

In the analysis of COMB we use the same notation as above. Let I be an instance of Δ-DECOM, and let Z be an optimal solution of I. We add (4), (7), and (8) to get

$$\begin{aligned} 3 \cdot \mathrm{COMB}(I) &\le \mathrm{TP}(I) + \mathrm{MA}(I) + \mathrm{MA}(I) \\ &\le 2 \sum_{(i,j,k)\in Z} (d_{ik} + d_{jk}) + 2 \sum_{(i,j,k)\in Z} (d_{ij} + d_{ik}) + 2 \sum_{(i,j,k)\in Z} (d_{ij} + d_{jk}) \\ &= 4 \sum_{(i,j,k)\in Z} (d_{ij} + d_{ik} + d_{jk}) \;=\; 4 \cdot \mathrm{OPT}(I). \end{aligned} \tag{10}$$

To see that equality may be attained in (10), consider the instance I depicted in Fig. 4. This instance has $|R| = 8$ and $|G| = 4$. All edges in the picture correspond to distance 1, and all non-edges in the picture correspond to distance 2. Observe the following: If a triple (i, j, k) forms a triangle in the picture (with all three edges present), then its cost is 3. If a triple does not form a triangle, then its cost is at least 4. Now an optimal solution consists of the four triangles (r_1, r_8, g_1), (r_2, r_3, g_2), (r_6, r_7, g_3), and (r_4, r_5, g_4), and the corresponding optimal cost is $\mathrm{OPT}(I) = 12$. How do our heuristics TP and MA behave on instance I?

- Suppose that the transportation problem in heuristic TP assigns r_5 and r_8 to g_1; assigns r_2 and r_7 to g_2; assigns r_3 and r_6 to g_3; and assigns r_1 and r_4 to g_4. Then each of the four resulting triples has cost 4, and hence $\mathrm{TP}(I) = 16$.
- Suppose that the first stage of heuristic MA finds the matching (r_1, r_2), (r_3, r_4), (r_5, r_6), and (r_7, r_8). Note that none of these four pairs belongs to any triangle in Fig. 4. Hence, no matter how the second stage matches the pairs to elements of G, every resulting triple will incur a cost of at least 4. This leads to $\mathrm{MA}(I) = 16$.

To summarize, the instance in Fig. 4 is a worst case instance for heuristic COMB with $\mathrm{COMB}(I) = (4/3)\,\mathrm{OPT}(I)$. We formulate the following theorem.

Theorem 3.4 *COMB is a polynomial time* $4/3$*-approximation algorithm for* Δ*-DECOM. Moreover, there exist instances I for which COMB*$(I) = (4/3)\,\mathrm{OPT}(I)$.

We point out that the idea of 'complementary' heuristics has been used before, see e.g. Friesen and Langston (1991) who give an example for the bin packing problem.

4 Computational experiments with heuristics TP and MA

In this section we report on our computational experiments with both heuristics TP and MA. Our primary goal is to see whether the quality of the solutions found by the heuristics differ when they are ran on different types of instances. To achieve this goal, we generated random instances of different types.

We consider two types of instances. Type 1 instances are instances where the elements of $R \cup G$ (in this section, the elements will be referred to as *points*) are uniformly generated in the two-dimensional plane, and the distance between a pair of points is the corresponding (rounded up) Euclidean distance. Type 2 instances are instances where the distance between each pair of points is in $\{1, 2\}$: with a certain probability p, the distance between i and j equals 1, and with probability $1 - p$, the distance between i and j equals 2, $i, j \in R \cup G$ (cf. Crama and Spieksma 1992). Concerning the type 1 instances, we consider three subtypes: type 1a, type 1b, and type 1c. For the instances of type 1a, the points in $R \cup G$ are generated randomly in a square of size $[0, 100] \times [0, 100]$. For the instances of type 1b, the points in R are generated in $[33, 100] \times [0, 100]$, while the points in G are generated in $[0, 32] \times [0, 100]$. For the instances of type 1c, the points in R are generated in $[0, 32] \times [0, 100]$, while the points in G are generated in $[33, 100] \times [0, 100]$. Concerning the type 2 instances, we consider two subtypes. For the instances of type 2a, the distance between any pair of points in $R \cup G$ equals 1 with probability $\frac{1}{n}$, and equals 2 with probability $1 - \frac{1}{n}$. For the instances of type 2b, the distance between any pair of points in R equals 2 with probability $\frac{1}{n}$, and equals 1 with probability $1 - \frac{1}{n}$, while the distance between a point in R and a point in G equals 1 with probability $\frac{1}{n}$, and equals 2 with probability $1 - \frac{1}{n}$ (recall that n is the number of green points in the instance).

For each of the five subtypes, we consider $n = 10$, $n = 20$, $n = 40$, $n = 60$ and $n = 80$. For each value of n we generate 10 instances. Thus, we have generated 250 instances in total; all instances can be found on http://www.econ.kuleuven.be/public/NDBAE53/2-1-AP.html.

All computations are done on a regular PC with AMD Turion (tm) 64 2 Mobile Technology TL-56 processor. Since both heuristics run very fast, we do not report running times.

Below, there is a table displaying the behavior of the two heuristics for each of the five subtypes. The format of such a table is as follows: the first column specifies the size of an instance. Next the second, third, and fourth column tabulate, for heuristic TP, the value found, the gap with the optimal value, and the number of instances (out of 10) for which TP finds an optimal solution. Columns five, six and seven display the same properties for heuristic MA. Finally, the last column gives the optimal value (optimum solutions have been found using CPLEX 12.1). Entries in the columns 'gap' are found by simply dividing the entry in the column 'value' by the entry in the column 'OPT'.

Table 2 Performance of MA and TP on instances of type 1a

Size	TP			MA			OPT
	Value	Gap	Optimal?	Value	Gap	Optimal?	
$n = 10$	670.4	1.074	1	632.8	1.014	5	624
$n = 20$	1,049.7	1.101	0	969.7	1.017	1	953.1
$n = 40$	1,689.8	1.122	0	1,516.7	1.008	1	1,505.4
$n = 60$	2,031	1.129	0	1,816.1	1.010	0	1,798.2
$n = 80$	2,457.3	1.121	0	2,212.4	1.009	0	2,192.9

Table 3 Performance of MA and TP on instances of type 1b

Size	TP			MA			OPT
	Value	Gap	Optimal?	Value	Gap	Optimal?	
$n = 10$	1,280.5	1.094	0	1,172.1	1.001	7	1,170.9
$n = 20$	2,558.4	1.138	0	2,247.6	1.000	8	2,247.4
$n = 40$	5,067.2	1.152	0	4,400.7	1.000	8	4,400.2
$n = 60$	7,636.9	1.165	0	6,556.9	1.000	6	6,556.3
$n = 80$	1,0092.4	1.173	0	8,603.7	1.000	0	8,602.3

Table 4 Performance of MA and TP on instances of type 1c

Size	TP			MA			OPT
	Value	Gap	Optimal?	Value	Gap	Optimal?	
$n = 10$	1,232.8	1.040	0	1,185.7	1.000	9	1,185.6
$n = 20$	2,403.6	1.066	0	2,254.8	1.000	10	2,254.8
$n = 40$	4,516.0	1.079	0	4,184.7	1.000	6	4,184.2
$n = 60$	7,107.5	1.081	0	6,577.9	1.000	2	6,576.4
$n = 80$	9,151.7	1.083	0	8,450.7	1.000	1	8,449.3

Notice that each of the entries in the second, third, fifth, sixth, and eighth column are averages over 10 instances.

Let us now discuss the results displayed in Tables 2, 3, 4, 5, 6. In general, the performance of the best of these two heuristics is quite good. Indeed, COMB (the best of TP and MA) finds, for each of the 250 instances, a solution within 6 % of the optimum value (and COMB is usually much better). Another general observation is that the performance of the heuristics only very mildly deteriorates with the size of the problem.

For the instances of type 1, MA, in line with the approximation ratios, performs very well: it consistently finds solutions that are within 1 % of the optimum. Solutions found by TP are worse; this can be explained by the fact that TP ignores distances between red points (red-red distances) completely. However, when the green-red distances are

Table 5 Performance of MA and TP on instances of type 2a

Size	TP			MA			OPT
	Value	Gap	Optimal?	Value	Gap	Optimal?	
$n = 10$	48.3	1.108	0	45.3	1.039	0	43.6
$n = 20$	94.1	1.102	0	87.9	1.029	0	85.4
$n = 40$	191.4	1.110	0	177.2	1.027	0	172.5
$n = 60$	286.1	1.110	0	265.6	1.031	0	257.7
$n = 80$	382.9	1.125	0	351.5	1.033	0	340.3

Table 6 Performance of MA and TP on instances of type 2b

Size	TP			MA			OPT
	Value	Gap	Optimal?	Value	Gap	Optimal?	
$n = 10$	38.4	1.021	5	41.2	1.096	0	37.6
$n = 20$	74.7	1.008	5	82.3	1.111	0	74.1
$n = 40$	152.0	1.003	6	167.5	1.106	0	151.5
$n = 60$	227.0	1.003	5	251.2	1.110	0	226.3
$n = 80$	306.1	1.001	7	337.3	1.103	0	305.8

comparable (as is the case for the instances of type 1b and type 1c), ignoring red-red distances becomes less problematic when the red points are less spread out. That explains why TP performs better on instances of type 1c than on instances of type 1b. Still, for geometric instances, MA is the superior heuristic.

For instances of type 2, the behavior of the heuristics is different. MA behaves worse on these instances compared to its performance on instances of type 1. However, for instances of type 2a (this is the case where almost all distances equal 2), MA still finds better solutions than TP does. The reason for this is that the red-red distances in TP's solutions equal 2 with high probability, whereas MA manages, in its first phase, to obtain short red-red distances. In subtype 2b, however, we have ensured that almost all red-red distances are 1. In such a case, the quality of the green-red assignment will determine the quality of the final solution. Therefore, TP finds the best solutions in this case.

5 Conclusion

Let us shortly elaborate on the question to what extent the approximation results from Sect. 3 can be generalized. The fact that Δ-DECOM is defined using two different colors seems instrumental for the analysis of heuristics TP and MA. Indeed, suppose we consider the problem where we are given $3n$ points, all of the same color (i.e., 3DM), together with "triangle-inequality-satisfying-distances" d, and the goal is to find n triples minimizing the corresponding triple costs. First, the heuristic TP is not easily generalized to this setting, and second, although heuristic MA can be generalized to this setting (in stage 1, we compute a minimum-weight matching on $2n$ out of the $3n$ points, and in stage 2, we assign the remaining n points to the pairs found in stage 1), it is not hard to find instances for which this generalization of MA performs arbitrarily bad.

Another generalization that is interesting to consider is the so-called p-to-q assignment problem; this is in fact a multi-index assignment problem (see Bandelt et al. 1994). In this problem we are given np green points, and nq red points, together with "triangle-inequality-satisfying-distances" d, and the goal is to find n $(p+q)$-tuples consisting of p green points and q red points while minimizing the corresponding clique costs (the clique costs are found by summing all distances within a $(p+q)$-tuple). Although one easily formulates the corresponding transportation problem (that forms the core of heuristic TP), it is not hard to exhibit instances for which the bound resulting from the solution of this transportation problem is arbitrarily bad compared to the optimal value, even for $p = q = 2$. On the positive side however, it is not hard to argue that heuristic TP is a tight p-approximation algorithm for the p-to-1 assignment problem.

Thus, possible avenues for further research include

- finding polynomial-time, constant factor approximation algorithms for the corresponding special case of 3DM (see Table 1),
- finding tight constant-factor approximation algorithms for the p-to-q assignment problem in case $p, q \geq 2$,
- investigating the case of Δ-DECOM where the points are embedded in the plane, and
- decreasing the gap between the currently known lower and upper bound of what is achievable (in terms of approximation) by polynomial time algorithms. More concrete: can we find approximation algorithms with a worst-case ratio less than 4/3? And/or can we improve the constant below which polynomial-time approximation algorithms cannot reach (unless P = NP)?

References

Bandelt H-J, Crama Y, Spieksma FCR (1994) Approximation algorithms for multidimensional assignment problems with decomposable costs. Discret Appl Math 49:25–50

Biyani P, Wu X, Sinha A (2005) Joint classification and pairing of human chromosomes. IEEE/ACM Trans Comput Biol Bioinf 2:102–109

Burkard RE, Rudolf R, Woeginger GJ (1996) Three-dimensional axial assignment problems with decomposable cost coefficients. Discret Appl Math 65:123–140

Crama Y, Spieksma FCR (1992) Approximation algorithms for three-dimensional assignment problems with triangle inequalities. Eur J Oper Res 60:273–279

Dutta A, Tsiotras P (2008) Egalitarian peer-to-peer satellite refueling strategy. J Spacecr Rocket 45:608–618

Friesen D, Langston M (1991) Analysis of a compound bin packing algorithm. SIAM J Discret Math 4:61–79

Goossens D, Polyakovskiy S, Spieksma FCR, Woeginger GJ (2010) The approximability of three-dimensional assignment problems with bottleneck objective. Optim Lett 4:7–16

Garey MR, Johnson DS (1979) Computers and intractability: a guide to the theory of NP-completeness. Freeman, San Francisco

Kann V (1991) Maximum bounded 3-dimensional matching is MAX SNP-complete. Inf Process Lett 37:27–35

Magyar G, Johnsson M, Nevalainen O (2000) An adaptive hybrid genetic algorithm for the three-matching problem. IEEE Trans Evol Comput 4:135–146

Petrank E (1994) The hardness of approximation: gap location. Comput Complex 4:133–157

Schrijver A (2003) Combinatorial optimization: polyhedra and efficiency. Springer, Berlin

Spieksma FCR, Woeginger GJ (1996) Geometric three-dimensional assignment problems. Eur J Oper Res 91:611–618

Urban T, Russel R (2003) Scheduling sport competitions on multiple venues. Eur J Oper Res 148:302–311

Wilson B (1963) Surf city. Liberty records