

# An experimental analysis of a GP hyperheuristic approach for evolving low-cost heuristics for profile reductions

L. M. Silva<sup>1</sup> and S. L. Gonzaga de Oliveira<sup>1</sup>

<sup>1</sup>Departamento de Ciência da Computação  
Universidade Federal de Lavras (UFLA) – Lavras, MG – Brazil

{sanderson, lsilva}@ufla.br

**Abstract.** Researchers used graph-theory approaches to design the state-of-the-art low-cost heuristics for profile reduction. This paper evolves and selects four low-cost heuristics for profile reduction using a genetic programming hyperheuristic approach. This paper evaluates the resulting heuristics for profile reduction from the genetic programming hyperheuristic approach in two application areas against the low-cost heuristics for solving the problem. The results obtained on a set of standard benchmark matrices taken from the SuiteSparse sparse matrix collection indicate that the resulting heuristics from the genetic programming hyperheuristic approach does not compare favorably with a high-quality heuristics for profile reduction.

## 1. Introduction

The solution of large-scale sparse linear systems  $Ax = b$ , where  $A = [a_{ij}]$  is an  $n \times n$  large-scale sparse matrix,  $x$  is the unknown  $n$ -vector solution and  $b$  is a known  $n$ -vector is relevant in several application areas in science and engineering, such as 2-D/3-D and thermal problems. The solution of large-scale sparse linear systems is usually the stage of the numerical simulation that requires the highest execution costs.

For the low-cost solution of large and sparse linear systems by iterative and direct methods, adequate graph labeling (of a graph corresponding to matrix  $A$ ) is advantageous. When using an appropriate graph labeling, the coefficient matrix  $A$  will have a small profile. For instance, when using iterative methods for solving linear systems, modern hierarchical memory architecture and paging policies favor programs that consider the locality of reference as a critical aspect. The use of a heuristic for profile reduction is an adequate choice to obtain a sequence of graph vertices with the spatial locality. Thus, numerical applications use heuristics for profile reduction to achieve low execution times for solving large sparse linear systems by iterative methods [Gonzaga de Oliveira et al. 2018a, Gonzaga de Oliveira et al. 2018b].

Heuristics for profile reduction place nonzero coefficients of a sparse matrix as close to the main diagonal as possible. Let  $A = [a_{ij}]$  be an  $n \times n$  symmetric adjacency matrix associated with an undirected graph  $G = (V, E)$  composed of a set of vertices  $V$  and a set of edges  $E$ . The profile of matrix  $A$  is defined as  $profile(A) = \sum_{i=1}^n \beta_i(A)$ , where  $\beta_i(A) = i - \min_{1 \leq j \leq i} [j : a_{ij} \neq 0]$ . Equivalently, the profile of  $G$  for a vertex labeling  $S = \{s(v_1), s(v_2), \dots, s(v_{|V|})\}$  (i.e. a bijective mapping from  $V$  to the set  $\{1, 2, \dots, |V|\}$ ) is  $profile(G) = \sum_{v \in V} \max_{\{v, u\} \in E} [|s(v) - s(u)|]$ . The profile minimization problem is  $\mathcal{NP}$ -hard [Lin and Yuan 1994].

An efficient solution of linear systems minimizes the total computing time, including the reordering time, at least when only a single linear system is to be solved. Thus, heuristics for profile reduction must be capable of obtaining high-quality profile results at a low cost. Previous publications [Bernardes and Gonzaga de Oliveira 2015, Gonzaga de Oliveira et al. 2018a, Gonzaga de Oliveira et al. 2018b] have reviewed several heuristics for profile reduction.

In this field, scientific and engineering applications apply low-cost heuristics for profile reductions when the computational time is a critical subject. Therefore, in the case of instances of rather large dimensions, a practical option is to use low-cost heuristics for obtaining satisfactory-quality solutions for problem instances defined by such matrices.

Koohestani and Poli [Koohestani and Poli 2015b, Koohestani and Poli 2015a] proposed a genetic programming system for reducing the profile size of sparse matrices. The genetic programming hyperheuristic approach evolved a variant of Sloan's algorithm [Sloan 1989]. The strategy included in the resulting heuristic components of the eigenvector corresponding to the first nonzero eigenvalue (Fiedler vector) of the Laplacian matrix associated with the matrix used in the learning process.

Koohestani and Poli [Koohestani and Poli 2015b] compared their genetic programming hyperheuristic approach for profile reduction with six (RCM [George 1971], Gibbs-King [Gibbs 1976, Lewis 1982], Sloan's [Sloan 1989], Spectral [Barnard et al. 1993], a modified hybrid algorithm that combines a spectral ordering with Sloan's algorithm [Reid and Scott 1999], and Sloan-MGPS [Reid and Scott 1999]) heuristics when applied to 34 small-scale matrices with sizes ranging from 59 to 5,300. Their faster-resulting heuristic is time-consuming. For example, both the faster-resulting heuristic proposed by Koohestani and Poli [Koohestani and Poli 2015b] and Sloan's algorithm [Sloan 1989] take approximately two seconds when applied in similar machines to matrices with sizes of 5,300 and 1.2 million, respectively. Thus, the faster approach proposed by Koohestani and Poli [Koohestani and Poli 2015b] is not practical for large-scale matrices.

Koohestani and Poli [Koohestani and Poli 2015a] compared their genetic programming hyperheuristic approach for profile reduction with the RCM [George 1971], Gibbs-King [Gibbs 1976, Lewis 1982], Spectral [Barnard et al. 1993], and Sloan's [Sloan 1989] heuristics when applied to 24 small-scale instances with sizes ranging from 59 to 2,680. The genetic programming hyperheuristic approach also performed poorly compared with these four algorithms concerning execution times. However, the authors did not compare their genetic programming hyperheuristic approach with promising heuristics for profile reducing, such as the MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], and Hu-Scott [Hu and Scott 2001] algorithms.

This paper presents a Genetic Programming Hyperheuristic (GPHH) for the matrix profile reduction problem. The work extends the approach presented by Koohestani and Poli [Koohestani and Poli 2015b, Koohestani and Poli 2015a] by the input of four heuristics, already known for the problem, to the GPHH system. Specifically, this approach evolves four low-cost heuristics for profile reduction (Sloan's [Sloan 1989], MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], and NMPG [Gonzaga de Oliveira et al. 2019]) using a genetic programming hyperheuristic

approach to obtain a low-cost heuristic for profile reduction. Since the objective is to find suitable weights for these heuristics, the computing time of the resulting heuristic is proportional to the CPU times of the algorithm evolved. Unlike the approach proposed by Koohestani and Poli [Koohestani and Poli 2015b, Koohestani and Poli 2015a], the genetic programming hyperheuristic strategy proposed in the present study either evolves or selects a heuristic for profile reduction for a specific application field.

The main novelty presented in this study regarding the approach developed by Koohestani and Poli [Koohestani and Poli 2015b, Koohestani and Poli 2015a] is that, as previously mentioned, we provide four heuristics for profile reduction as input to the genetic programming hyperheuristic approach. Furthermore, the genetic programming hyperheuristic approach evolves the original algorithm into a low-cost heuristic for profile reduction or selects the original heuristic to reduce the profile of matrices arising from a specific application area.

This paper evaluates the resulting heuristics for profile reduction in two application fields against Sloan’s [Sloan 1989], MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], Sloan-MGPS [Reid and Scott 1999], Hu-Scott [Hu and Scott 2001], and NMPG [Gonzaga de Oliveira et al. 2019] heuristics.

The remainder of this paper is organized as follows. Section 2 reports the related work in this field. Section 3 introduces the proposed genetic programming hyperheuristic approach for profile reduction. Section 4 describes how the experiments were conducted. Section 5 discusses the results. Finally, section 6 addresses the conclusions.

## 2. Heuristics evaluated

Practitioners have proposed a considerable number of heuristics for reducing the profile of sparse matrices [Bernardes and Gonzaga de Oliveira 2015]. The state-of-the-art low-cost heuristics for profile reduction are based on graph-theory concepts [Gonzaga de Oliveira et al. 2018a, Gonzaga de Oliveira et al. 2018b].

Sloan [Sloan 1989] proposed one of the most important heuristics in this field. This heuristic is still one of the most widely used reordering algorithm for reducing the profile size of sparse matrices (e.g., [Bernardes and Gonzaga de Oliveira 2015, Kumfert and Pothen 1997, Medeiros et al. 1993, Reid and Scott 1999, Gonzaga de Oliveira et al. 2018a, Gonzaga de Oliveira et al. 2018b, Gonzaga de Oliveira et al. 2019]). The reason is that it is inexpensive (in terms of execution times and storage costs) and generates high-quality solutions. Sloan’s heuristic [Sloan 1989] uses two weights in its priority scheme in order to label the vertices of the instance:  $w_1$ , associated with the distance  $d(v, e)$  from the vertex  $v$  to a pseudo-peripheral (target end) vertex  $e$  that belongs to the last level of the level structure rooted at a starting vertex  $s$ , and  $w_2$ , associated with the degree of each vertex. The priority function  $p(v) = w_1 \cdot d(v, e) - w_2 \cdot (deg(v) + 1)$  employed in Sloan’s heuristic [Sloan 1989] presents different scales for both criteria. The value of  $deg(v) + 1$  ranges from 1 to  $m + 1$  [where  $m = \max_{v \in V} [deg(v)]$  is the maximum degree found in the graph  $G = (V, E)$ ], and  $d(v, e)$  ranges from 0 (when  $v = e$ ) to the eccentricity  $\ell(e)$  (of the target end vertex  $e$ ).

The MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], and Sloan-

MGPS [Reid and Scott 1999] heuristics are based on Sloan’s heuristic [Sloan 1989]. Specifically, the Sloan-MGPS heuristic [Reid and Scott 1999] is essentially the Sloan’s heuristic [Sloan 1989] with the starting and target end vertices provided by an algorithm named modified GPS [Reid and Scott 1999].

The MPG heuristic [Medeiros et al. 1993] employs two max-priority queues:  $t$  contains candidate vertices to be labeled, and  $q$  contains vertices belonging to  $t$  and also vertices that can be inserted to  $t$ . Similarly to Sloan’s heuristic [Sloan 1989] (and its variations), the current degree of a vertex is the number of adjacencies to vertices that neither have been labeled nor belong to  $q$ . A main loop performs three steps. First, a vertex  $v$  is inserted into  $q$  to maximize a specific priority function. Second, the current degree  $cdeg(v)$  of each vertex  $v \in t$  is observed: the algorithm labels a vertex  $v$  if  $cdeg(v) = 0$ , and the algorithm removes from  $t$  a vertex  $v$  (i.e.,  $t \leftarrow t - \{v\}$ ) if  $cdeg(v) > 1$ . Third, if  $t$  is empty, the algorithm inserts into  $t$  each vertex  $u \in q$  with priority  $p_u \geq p_{max}(q) - 1$  where  $p_{max}(q)$  returns the maximum priority among the vertices in  $q$ . The priority function in the MPG heuristic is  $p(v) = d(v, e) - 2 \cdot cdeg(v)$ .

Kumfert and Pothen [Kumfert and Pothen 1997] normalized the two criteria used in Sloan’s algorithm with the objective of proposing the Normalized Sloan (NSloan) heuristic [Kumfert and Pothen 1997]. These authors used the priority function  $p(v) = w_1 \cdot d(v, e) - w_2 \cdot \lfloor d(s, e)/m \rfloor \cdot (deg(v) + 1)$ . In addition, a recent publication normalized the two criteria used in the MPG heuristic aiming at proposing the NMPG heuristic [Gonzaga de Oliveira et al. 2019].

Regarding Sloan’s [Sloan 1989], NSloan [Kumfert and Pothen 1997], and Sloan-MGPS [Reid and Scott 1999] heuristics, this study set the two weights as described in the original papers. When the original publications recommended more than one pair of values, a previous publication performed exploratory investigations to determine the pair of values that obtain the best profile results [Gonzaga de Oliveira et al. 2018a]. Thus, the two weights are set as  $w_1 = 1$  and  $w_2 = 2$  for Sloan’s and Sloan-MGPS [Reid and Scott 1999] heuristics, and as  $w_1 = 2$  and  $w_2 = 1$  for the NSloan heuristic [Kumfert and Pothen 1997]. The Hu-Scott heuristic [Hu and Scott 2001] is a multilevel algorithm that uses a maximal independent vertex set for coarsening the adjacency graph of the matrix and employs the Sloan-MGPS heuristic [Reid and Scott 1999] on the coarsest graph.

### 3. A genetic programming hyperheuristic approach for profile reduction of symmetric matrices

This section presents a genetic programming hyperheuristic approach for profile reduction. We followed the main steps of the genetic programming metaheuristic [Koza 1992] to design the genetic programming hyperheuristic approach for profile reduction.

We established the genetic programming hyperheuristic with the central structure of Sloan’s [Sloan 1989], MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], and NMPG [Gonzaga de Oliveira et al. 2019] heuristics. Furthermore, the hyperheuristic approach generates random weights for the priority formula employed in these four heuristics. We implemented the codes in the C++ programming language.

Individuals in the genetic programming system is a set of three components: the base heuristics to be used and its two weights. The initial population is randomly generated, except for four individuals, which are precisely the original weights (1 and 2) of Sloan’s, NSloan, MPG, and NMPG heuristics. The fitness function employed in the genetic programming hyperheuristic approach is the sum of the difference between the original profile and the profile of the labeling produced by the heuristic associated with the individual program in each instance used in the learning process. Thus, the objective is to maximize the fitness value.

Due to the high-quality results obtained by Koohestani and Poli [Koohestani and Poli 2015b, Koohestani and Poli 2015a], we used the same parameters employed by the authors. Column GPHH<sub>1</sub> in Table 1 shows the parameters used in this study. In addition, column GPHH<sub>2</sub> in Table 1 shows new parameters used by the GPHH approach.

**Tabela 1. Parameters used in the experiments with the GPHH approach.**

| Parameter                     | GPHH <sub>1</sub> | GPHH <sub>2</sub> |
|-------------------------------|-------------------|-------------------|
| Maximum number of generations | 100               | 120               |
| Population size               | 1000              | 1250              |
| Tournament size               | 3                 | 4                 |
| Elitism rate                  | 0.1%              | 0.08%             |
| Reproduction rate             | 0.9%              | 0.92%             |
| Crossover rate                | 80%               | 0.84%             |
| Mutation rate                 | 19%               | 15%               |

The GPHH approach employs tournament selection based on fitness to choose the individual program(s) from the population to participate in genetic operations. The GPHH approach applies genetic procedures of reproduction, crossover, and point mutation with specified probabilities to create new individual programs. The system also uses elitism to ensure that the best individual program in one generation is transferred unaltered to the next generation. The GPHH system selects the best program tree that appears in the last generation as the final result of a run. Algorithm 1 shows the GPHH system, recalling that Table 1 shows its parameters.

The GPHH system creates an initial population [i.e., individual programs associated with a heuristic for profile reduction (Sloan, NSloan, MPG, or NMPG) and its weights] at line 2. The loop in lines 4–13 evaluates each (individual) program of the population. The loop in lines 5–10 applies the heuristic for profile reduction associated with the individual program to each matrix contained in the set *Instances* (see line 6). The GPHH approach transfers only one individual program to the new generation in line 12 because of the number of individual programs and the elitism rate used. Thus, the GPHH<sub>1</sub> (GPHH<sub>2</sub>) approach used 1,000 (1,250) individual programs and set the elitism rate as 0.1 (0.08)%.

The GPHH system creates a new population in line 12 using the parameters of crossover, mutation, and reproduction rates. The genetic programming hyperheuristic returns the best individual program at line 14.

**Input:** a set of matrices *Instances*, a set of parameters:  
*maximumNumberOfGenerations*, *populationSize*, *tournamentSize*,  
*reproductionRate*, *crossoverRate*, *mutationRate*,  
*mutationPerNodeRate*.

**Output:** an evolved heuristic for profile reduction.

```

1 begin
2   Population  $\leftarrow$  create an initial population with populationSize individual
   programs;
3    $k \leftarrow 0$ ;
4   while ( $k < \text{maximumNumberOfGenerations}$ ) do
5     for (each individual  $\in$  Population) do
6       for (each matrix  $i \in$  Instances) do
7         apply the heuristic associated with individual to matrix  $i$ ;
8         newProfile  $\leftarrow$  profile of the new labeling of matrix  $i$ ;
9         diff  $\leftarrow$  original profile subtracted from newProfile;
10        sum diff to the total reduction profile stored in individual;
11       $k \leftarrow k + 1$ ;
12      Population  $\leftarrow$  create a new Population using tournamentSize,
       reproductionRate, crossoverRate, mutationRate,
       mutationPerNodeRate, and including the best individual program to
       the new population (elitism);
13      delete the old population of individual program;
14  return the best individual program;
```

**Algorithm 1:** A genetic programming hyperheuristic for profile reduction.

We used real symmetric matrices arising from 2-D/3-D and thermal problems contained in the SuiteSparse matrix collection [Davis and Hu 2011] in the learning process. Table 2 shows the instances used in the learning process.

As a result of the learning process, the GPHH<sub>1</sub> approach generated the weights 0.324075 and 0.8199 (0.524625 and 0.8994) for the MPG heuristic [Medeiros et al. 1993] to apply the resulting heuristic to instances originating from 2-D/3-D (thermal) problems. The GPHH<sub>2</sub> approach generated the weights 0.30923 and 0.821214 (0.324075 and 0.819900) for Sloan's [Sloan 1989] (NSloan [Kumfert and Pothen 1997]) algorithm to apply the resulting heuristic to instances originating from 2-D/3-D (thermal) problems.

**Tabela 2. Matrices used in the learning process for the genetic programming hyperheuristic.**

| Problem     | Matrices and their size |          |           |                |          |          | Training days |
|-------------|-------------------------|----------|-----------|----------------|----------|----------|---------------|
| 2-D/<br>3-D | Matrix                  | mario001 | wathen120 | nd12k          | aug3dcqp | helm3d01 | 8             |
|             | $n$                     | 38,434   | 36,441    | 36,000         | 35,543   | 32,226   |               |
|             | $ E $                   | 204,912  | 565,761   | 14,220,946     | 128,115  | 428,444  |               |
| thermal     | Matrix                  | ted_B    |           | ted_B_unscaled |          |          | 5             |
|             | $n$                     | 10,605   |           | 10,605         |          |          |               |
|             | $ E $                   | 144,579  |           | 144,579        |          |          |               |

We also evaluated the heuristics with the Hager exchange method [Hager 2002, Reid and Scott 2002]. This algorithm refines a profile-reducing permutation of a symmetric matrix. In this case, the GPHH<sub>2</sub> approach was evaluated. The GPHH<sub>2</sub> approach generated the weights 0.32643 and 0.872345 (0.441254 and 0.634901) for the NSloan [Kumfert and Pothen 1997] (MPG [Medeiros et al. 1993]) heuristic to apply the resulting heuristic to instances originating from 2-D/3-D (thermal) problems.

#### 4. Description of the tests

The Hu-Scott heuristic [Hu and Scott 2001] is a high-quality heuristic for profile reduction. We used the Hu-Scott (MC73 routine) and Sloan-MGPS (MC60 routine) heuristics contained in the HSL [STFC 2018]. We employed the Hager exchange algorithm, namely the MC67 routine, contained in the same mathematical software library. We employed the Fortran programming language to use these routines.

This appraisal also used the C++ programming language to implement the other low-cost heuristics for profile reduction evaluated. The implementations of the heuristics for profile reductions appraised in this study employ binary heaps to code the priority queues (although the original Sloan's algorithm [Sloan 1989] used a linked list in its implementation). A previous publication [Gonzaga de Oliveira et al. 2018a] presented the testing and calibration performed to compare the codes with the ones used by the original proposers of the low-cost heuristics (Sloan's [Sloan 1989], MPG [Medeiros et al. 1993], NSloan [Kumfert and Pothen 1997], and Sloan-MGPS [Reid and Scott 1999]) to ensure that the codes employed in the present study were comparable to the formerly proposed algorithms.

The experiments used eight real symmetric matrices taken from the SuiteSparse matrix collection [Davis and Hu 2011]. We used the four largest real symmetric matrices arising from 2-D/3-D and thermal problems contained in this matrix collection.

The workstations used in the execution of the simulations featured an Intel® Core™ i7-4770 (CPU 3.4 GHz, 8 MB Cache, 8 GB of main memory DDR3 1333 MHz) (Intel; Santa Clara, CA, United States). The profile reduction depends on the choice of the initial ordering, and this paper considers the original ordering given in the matrix.

#### 5. Results and analysis

This section shows the results of several low-cost heuristics for profile reductions when applied to eight matrices with sizes ranging from 102,158 to 1,228,045 [up to 8,580,313 nonzero coefficients (nzc)]. Tables 3 and 4 show the name, size, and original profile ( $P_0$ ) of the matrices used in this computational experiment. Also, these tables show the profile results and CPU times obtained by Sloan's, MPG, NSloan, Sloan-MGPS, Hu-Scott, NMPG, and the resulting heuristics from the GPHH approach when applied to eight matrices arising from 2-D/3-D and thermal problems, respectively. The same tables also show the profile results yielded by these heuristics when applied with the Hager exchange algorithm.

Table 3 shows that the Hu-Scott heuristic achieved the best profile results when applied to the two smallest matrices (darcy003, helm2d03) arising from 2-D/3-D problems. On the other hand, the heuristic yielded unsatisfactory profile results when applied to

**Tabela 3. Profile results yielded by several heuristics when applied to four large-scale matrices arising from 2-D/3-D problems.**

| Matrix   | $n$       | nzc       | $P_0$         | Heuristic         | Profile     | $t(s)$ | Hager              |
|----------|-----------|-----------|---------------|-------------------|-------------|--------|--------------------|
| darcy003 | 389,874   | 2,097,566 | 778,956,927   | Hu-Scott          | 95,382,099  | 3.1    | 80,653,654         |
|          |           |           |               | GPHH <sub>1</sub> | 114,701,583 | 9.4    | —                  |
|          |           |           |               | GPHH <sub>2</sub> | 114,956,190 | 9.4    | 94,635,636         |
|          |           |           |               | NMPG              | 115,043,267 | 4.1    | 95,281,674         |
|          |           |           |               | MPG               | 115,231,979 | 4.0    | 95,317,875         |
|          |           |           |               | Sloan             | 115,562,260 | 0.4    | 95,465,477         |
|          |           |           |               | NSloan            | 115,660,780 | 0.4    | 95,654,660         |
|          |           |           |               | Sloan-MGPS        | 115,746,306 | 0.4    | 95,436,786         |
| helm2d03 | 392,257   | 2,741,935 | 1,334,787,533 | Hu-Scott          | 204,823,797 | 1.4    | 177,536,560        |
|          |           |           |               | MPG               | 211,075,951 | 0.5    | 191,166,860        |
|          |           |           |               | NMPG              | 211,101,243 | 0.5    | 191,264,028        |
|          |           |           |               | GPHH <sub>1</sub> | 219,484,646 | 0.5    | —                  |
|          |           |           |               | GPHH <sub>2</sub> | 220,231,204 | 0.5    | 198,676,364        |
|          |           |           |               | NSloan            | 226,788,690 | 0.5    | 206,979,645        |
|          |           |           |               | Sloan-MGPS        | 241,992,888 | 0.4    | 224,275,665        |
|          |           |           |               | Sloan             | 247,135,896 | 0.5    | 223,427,456        |
| ecology2 | 999,999   | 4,995,991 | 998,998,001   | NSloan            | 667,163,760 | 1.2    | 636,224,671        |
|          |           |           |               | GPHH <sub>2</sub> | 667,163,809 | 1.5    | 637,055,155        |
|          |           |           |               | Sloan-MGPS        | 667,163,817 | 1.3    | 636,272,926        |
|          |           |           |               | Sloan             | 667,163,920 | 1.2    | 636,272,831        |
|          |           |           |               | MPG               | 667,163,923 | 1.7    | 636,254,831        |
|          |           |           |               | NMPG              | 667,163,923 | 1.7    | <b>635,223,494</b> |
|          |           |           |               | GPHH <sub>1</sub> | 667,164,046 | 1.5    | —                  |
|          |           |           |               | Hu-Scott          | 924,354,061 | 2.1    | 823,445,152        |
| ecology1 | 1,000,000 | 4,996,000 | 999,000,999   | Sloan             | 667,165,500 | 1.2    | 634,354,424        |
|          |           |           |               | NSloan            | 667,165,500 | 1.3    | 634,354,424        |
|          |           |           |               | Sloan-MGPS        | 667,165,500 | 1.4    | 634,354,424        |
|          |           |           |               | GPHH              | 667,165,500 | 1.5    | 634,354,424        |
|          |           |           |               | MPG               | 667,165,500 | 1.7    | 634,354,424        |
|          |           |           |               | NMPG              | 667,165,500 | 1.7    | 634,354,424        |
|          |           |           |               | Hu-Scott          | 936,696,940 | 2.0    | 845,787,851        |

the two highest instances originating from this application area. Table 3 shows that the NSloan heuristic delivered the best profile result when applied to the instance ecology2 (composed of 999,999 vertices and 4,995,991 edges) originating from the same application area. Sloan's algorithm obtained the best profile result at lower execution times than the other heuristics evaluated when applied to the instance ecology1 (composed of one million vertices and almost five million edges) originating from 2-D/3-D problems. Both GPHH<sub>1</sub> and GPHH<sub>2</sub> heuristics yielded the same profile result when applied to the matrix ecology1. The Hager exchange method favored the NMPG heuristic with the matrix ecology2 as input.



Table 4 shows that the Hu-Scott heuristic yielded the best profile results when applied to the four matrices arising from thermal problems. Both GPHH<sub>1</sub> and GPHH<sub>2</sub> heuristics yielded the same profile results when applied to these four matrices. The Hager exchange method favored Sloan’s algorithm when applied to the matrix thermomech\_dM.

The resulting heuristics from the GPHH approach did not yield better results than did the other low-cost heuristics for profile reduction evaluated when applied to matrices arising from two application areas. Thus, we did not use instances arising from other domain fields in the computational experiment.

**Tabela 4. Profile results delivered by several heuristics when applied to four large-scale matrices arising from thermal problems.**

| Matrix            | $n$       | nzc       | $P_0$         | Heuristic  | Profile     | $t(s)$ | Hager             |
|-------------------|-----------|-----------|---------------|------------|-------------|--------|-------------------|
| thermo<br>mech_TC | 102,158   | 711,558   | 2,667,823,445 | Hu-Scott   | 13,477,995  | 0.6    | 10,245,895        |
|                   |           |           |               | GPHH       | 15,329,403  | 0.2    | 12,428,512        |
|                   |           |           |               | NMPG       | 15,408,417  | 0.2    | 12,480,345        |
|                   |           |           |               | MPG        | 15,412,663  | 0.2    | 12,501,474        |
|                   |           |           |               | NSloan     | 15,446,340  | 0.2    | 12,236,432        |
|                   |           |           |               | Sloan      | 16,021,130  | 0.2    | 13,212,121        |
|                   |           |           |               | Sloan-MGPS | 16,109,532  | 0.2    | 13,218,441        |
| thermo<br>mech_TK | 102,158   | 711,558   | 2,667,823,445 | Hu-Scott   | 13,372,590  | 0.4    | 13,372,590        |
|                   |           |           |               | GPHH       | 15,329,403  | 0.2    | 15,329,403        |
|                   |           |           |               | NSloan     | 15,493,278  | 0.2    | 15,493,278        |
|                   |           |           |               | MPG        | 15,504,258  | 0.2    | 15,504,258        |
|                   |           |           |               | NMPG       | 15,513,164  | 0.2    | 15,513,164        |
|                   |           |           |               | Sloan      | 15,906,569  | 0.2    | 15,906,569        |
|                   |           |           |               | Sloan-MGPS | 16,109,532  | 0.2    | 16,109,532        |
| thermo<br>mech_dM | 204,316   | 1,423,116 | 2,081,359,188 | Hu-Scott   | 28,549,159  | 0.7    | 26,979,479        |
|                   |           |           |               | GPHH       | 30,811,966  | 0.4    | 28,623,743        |
|                   |           |           |               | NMPG       | 30,824,941  | 0.4    | 28,624,831        |
|                   |           |           |               | MPG        | 30,825,327  | 0.4    | 28,632,316        |
|                   |           |           |               | NSloan     | 30,939,618  | 0.3    | 28,738,717        |
|                   |           |           |               | Sloan      | 31,984,980  | 0.3    | <b>22,895,892</b> |
|                   |           |           |               | Sloan-MGPS | 32,276,780  | 0.3    | 30,366,662        |
| thermal2          | 1,228,045 | 8,580,313 | 2,117,727,528 | Hu-Scott   | 577,408,773 | 3.2    | 573,327,864       |
|                   |           |           |               | NMPG       | 583,761,864 | 15.4   | 579,799,820       |
|                   |           |           |               | MPG        | 583,834,354 | 15.3   | 579,945,450       |
|                   |           |           |               | NSloan     | 587,246,981 | 1.7    | 584,355,892       |
|                   |           |           |               | GPHH       | 587,662,601 | 17.0   | 585,573,510       |
|                   |           |           |               | Sloan      | 593,981,740 | 1.7    | 588,892,832       |
|                   |           |           |               | Sloan-MGPS | 609,179,419 | 1.7    | 594,358,318       |

## 6. Conclusions

This paper compared heuristics for profile reduction evolved by a genetic programming hyperheuristic approach with low-cost algorithms for this problem. The resulting heu-

ristics generated by the GPHH approach did not compare favorably with the Hu-Scott heuristic. The experiments conducted in this study relied on eight matrices arising from two application domains.

We intend to design new hyperheuristics based on different metaheuristics. A previous publication reported a description of parallel algorithms in this field [Gonzaga de Oliveira et al. 2018b]. Concerning massively-parallel SIMD processing, the next step of this work is to evaluate the low-cost heuristics for profile reductions implemented using parallel libraries (e.g., OpenMP, Galois, and Message Passing Interface systems) and in GPU-accelerated computing. Similarly, regarding massively parallel computing, an evaluation of these heuristics for profile reduction implemented within Intel<sup>®</sup> Math Kernel Library running on Intel<sup>®</sup> Xeon<sup>®</sup> Many Integrated Core Architecture, Scalable and Cascade processors is another future step of this investigation.

## Acknowledgment

The Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) supported the development of this work.

## Referências

- Barnard, S. T., Pothen, A., and Simon, H. D. (1993). A spectral algorithm for envelope reduction of sparse matrices. In *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*, pages 493–502, Portland, OR. ACM.
- Bernardes, J. A. B. and Gonzaga de Oliveira, S. L. (2015). A systematic review of heuristics for profile reduction of symmetric matrices. *Procedia Computer Science*, 51:221–230.
- Davis, T. A. and Hu, Y. (2011). The University of Florida sparse matrix collection. *ACM Transactions on Mathematical Software*, 38(1):1–25.
- George, A. (1971). *Computer implementation of the finite element method*. PhD thesis, Stanford University, Stanford, USA.
- Gibbs, N. (1976). A hybrid profile reduction algorithm. *ACM Transactions on Mathematical Software*, 2(4):378–387.
- Gonzaga de Oliveira, S. L., Bernardes, J. A. B., and Chagas, G. O. (2018a). An evaluation of low-cost heuristics for matrix bandwidth and profile reductions. *Computational & Applied Mathematics*, 37(2):1412–1471.
- Gonzaga de Oliveira, S. L., Bernardes, J. A. B., and Chagas, G. O. (2018b). An evaluation of reordering algorithms to reduce the computational cost of the incomplete cholesky-conjugate gradient method. *Computational & Applied Mathematics*, 37(3):2965–3004.
- Gonzaga de Oliveira, S. L., Osthoff, C., and Henderson, L. N. (2019). An Experimental Analysis of Heuristics for Profile Reduction. In *Proceedings of 18th International Conference on Computational Science and Its Applications, ICCSA. Lecture Notes in Computer Science book series (LNCS) vol. 11619*, pages 25–36, Saint Petersburg, Russia. Springer International Publishing.
- Hager, W. W. (2002). Minimizing the profile for a symmetric matrix. *SIAM Journal on Scientific Computing*, 23(5):1799–1816.

- Hu, Y. and Scott, J. A. (2001). A multilevel algorithm for wavefront reduction. *SIAM Journal on Scientific Computing*, 23(4):1352–1375.
- Koohestani, B. and Poli, R. (2015a). Addressing the envelope reduction of sparse matrices using a genetic programming system. *Computational Optimization and Applications*, 60(3):789–814.
- Koohestani, B. and Poli, R. (2015b). Evolving an improved algorithm for envelope reduction using a hyper-heuristic approach. *IEEE Transactions on Evolutionary Computation*, 18(4):543–558.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.
- Kumfert, G. and Pothén, A. (1997). Two improved algorithms for envelope and wavefront reduction. *BIT Numerical Mathematics*, 37(3):559–590.
- Lewis, J. G. (1982). Implementation of the Gibbs-Poole-Stockmeyer and Gibbs-King algorithms. *ACM Transactions on Mathematical Software*, 8(2):180–189.
- Lin, Y. X. and Yuan, J. J. (1994). Profile minimization problem for matrices and graphs. *Acta Mathematicae Applicatae Sinica*, 10(1):107–122.
- Medeiros, S. R. P., Pimenta, P. M., and Goldenberg, P. (1993). Algorithm for profile and wavefront reduction of sparse matrices with a symmetric structure. *Engineering Computations*, 10(3):257–266.
- Reid, J. K. and Scott, J. A. (1999). Ordering symmetric sparse matrices for small profile and wavefront. *International Journal for Numerical Methods in Engineering*, 45(12):1737–1755.
- Reid, J. K. and Scott, J. A. (2002). Implementing Hager’s exchange methods for matrix profile reduction. *ACM Trans. Math. Softw.*, 28(4):377–391.
- Sloan, S. W. (1989). A Fortran program for profile and wavefront reduction. *International Journal for Numerical Methods in Engineering*, 28(11):2651–2679.
- STFC (Accessed: December, 2018). The Science and Technology Facilities Council. HSL. A collection of Fortran codes for large scale scientific computation. <http://www.hsl.rl.ac.uk>.