Heuristic solutions for the (α, β)-k feature set problem

Leila M. Naeni[*] Amir Salehipour[†]

Abstract

The (α, β)-k Feature Set Problem (FSP) is a combinatorial optimization-based approach for selecting features. The (α, β)-k FSP selects a set of features such that the set maximizes the similarities between entities of the same group and the differences between entities of different groups. This study develops two heuristic algorithms for the (α, β)-k FSP. We tested the algorithms on 11 real-world instances ranging from medium to large. The computational results demonstrate the proposed heuristics compete well against the standard solver CPLEX.

Keywords: heuristics; feature selection; combinatorial optimization; data analysis

1 Introduction

The (α, β)-k Feature Set Problem (FSP) is a combinatorial optimization-based approach proposed in 2004 for the feature selection (Cotta et al., 2004). The (α, β)-k FSP aims to select the minimum number of features in order to distinguish two groups (classes) of data such that the selected set of features maximizes the similarities between entities of the same group and the differences between entities of different groups. The problem is a generalization of the k-Feature Set Problem, which is proven $\mathcal{NP}$-Hard (Cotta et al., 2004). Hence, the (α, β)-k FSP is also NP-Hard.

Generally speaking, feature selection, also known as variable selection, attribute selection or variable subset selection, is to choose a subset of features, out of a set of candidate features, such that the selected set best represents the whole in a particular aspect. Removing irrelevant or redundant features, and reducing the dimensionality of the dataset are two reasons to perform feature selection (Paula, 2012). These criteria are both interesting and important because given the size of the datasets we encounter in many applications, they ease analysis, utilization, and interpretation of high-dimensional datasets. One such example is the studies by Inostroza-Ponta et al. (2008); Inostroza-Ponta et al. (2011). The authors modeled a visualization problem as a Quadratic Assignment Problem.

Feature selection has a broad range of applications including machine learning and prediction, and in a variety of domains such as urban transport network planning (Ferchichi et al., 2009), stock price prediction (Meiri and Zahavi, 2006; Tsai and Hsiao, 2010), and computational biology and bioinformatics (Albrecht, 2006; Ravetti and Moscato, 2008; Ravetti et al., 2009; Ravetti et al., 2010; Fan and Chaovalitwongse, 2010; Paula et al., 2011; Haque et al., 2016).

There are several drawbacks and limitations in the available solution methods of the (α, β)-k FSP. The major limitation of the exact methods is that they are unable to solve medium and large instances (see for example Cotta et al. (2004); Berretta et al. (2008)), and not to mention that many applications

[*]School of Built Environment, University of Technology Sydney, NSW, Australia. Email: leila.mosleminaeni@uts.edu.au

[†]Corresponding author. School of Mathematical and Physical Sciences, University of Technology Sydney, NSW, Australia. Email: amir.salehipour@uts.edu.au

of (α, β)-k FSP involve large datasets. On the heuristic side, the algorithms of Paula (2012) benefit from general and randomized local searches developed for the traditional combinatorial optimization problems. Moreover, the algorithms were not tested on large instances. The present study is motivated by the computational difficulty of the (α, β)-k FSP, and lack of efficient solution methods, as well as broad range of applications of (α, β)-k FSP. The major contributions of the present study can be summarized as follow: (1) developing mathematical properties for the (α, β)-k FSP, and (2) proposing heuristic solutions in order to efficiently solve medium and large instances of the problem.

The remaining of this paper is organized as follows. Section 2 defines the (α, β)-k FSP. Several mathematical properties of the (α, β)-k FSP are discussed in Section 3. Section 4 develops two heuristic algorithms for solving the (α, β)-k FSP. Computational results are discussed in Section 5. Finally, the paper concludes with the outcomes of the study and a few research directions.

2 Problem statement

Assume two groups (classes) of data exist, for example, group 1 and group 2, and a set $J = \{1, \ldots, n\}, |J| = n$ of features each with a profile $P_j, \forall j \in J$ $(P = \{P_j\}, \forall j \in J)$ is given. A feature profile P_j includes a set of discrete values of either 0 or 1. Furthermore, let S_1 and S_2 denote the set of all entities in group 1 and group 2, where $S_1 = \{s_{11}, \ldots, s_{1n_1}\}, |S_1| = n_1$, and $S_2 = \{s_{21}, \ldots, s_{2n_2}\}, |S_2| = n_2$. Let I_1 and I_2 represent sets of pairs of entities of different groups, and of the same group. Then I_1 includes all pairs of entities (every combination of size two of entities) belonging to different groups, and I_2 includes all pairs of entities belonging to the same group. Sets I_1 and I_2 can be formed by using Equations (1) and (2).

$$I_1 = \{(s_{11}, s_{21}), \ldots, (s_{11}, s_{2n_2}), \ldots, (s_{1n_1}, s_{2n_2})\} \tag{1}$$

I_1 is the set of all pairs of entities $(s_{1t}, s_{2t'})$, where $s_{1t} \in S_1, \forall t = 1, \ldots, n_1$, and $s_{2t'} \in S_2, \forall t' = 1, \ldots, n_2$.

$$I_2 = \{(s_{11}, s_{12}), \ldots, (s_{11}, s_{1n_1}), \ldots, (s_{21}, s_{22}), \ldots, (s_{21}, s_{2n_2})\} \tag{2}$$

Similarly, I_2 includes all pairs of entities $(s_{1t}, s_{1t'})$, where $(s_{1t}, s_{1t'}) \in S_1, \forall t, t' = 1, \ldots, n_1, t \neq t'$, and $(s_{2t}, s_{2t'}) \in S_2$, where $(s_{2t}, s_{2t'}) \in S_2, \forall t, t' = 1, \ldots, n_2, t \neq t'$.

Given these definitions and notations, the (α, β)-k FSP is defined with three positive integer parameters α, β, and k. The value of α represents the minimum number of features that must explain the differences between any pair of entities of different groups. The value of β represents the minimum number of features that must explain the similarities between any pair of entities of the same group. Finally, k represents the number of features to be selected. More precisely, the (α, β)-k FSP has the following characteristics: (1) every element in I_1 must be "explained" (*covered*) by at least α features, (2) a set $J^* \subseteq J$ of features with the minimum cardinality, among all alternative sets, must be selected, and (3) every element in I_2 must be "explained" (covered) by at least β features, where $1 \leq \beta \leq \beta^*, \beta \in \mathbb{Z}^+$, and β^* is the maximum value of β.

Let us explain how we can build an instance of the (α, β)-k FSP from a dataset with two groups of data. Table 1 shows a dataset that includes two groups of data. Group 1 consists of three healthy samples (or entities), and group 2 consists of three disease samples (the number of entities in the groups do not need to be equal). The last row in Table 1 states the label of groups. Furthermore, the dataset includes five features, which may represent genes, probes, etc. The entities of Table 1 may refer to discretized gene expression levels. Here, a feature may be up-regulated (associated with

Table 1: The dataset with two groups (classes) of data.

Feature	Sample 1	Sample 2	Sample 3	Sample 4	Sample 5	Sample 6
A	0	0	0	0	1	0
B	1	1	1	1	0	1
C	1	1	1	0	1	0
D	1	1	1	1	0	0
E	0	0	1	0	0	0
Group	1	1	1	2	2	2

a value of 1) or down-regulated (associated with a value of 0) in a sample.

Applying Equations (1) and (2) results in $I_1 = \{(1,4),(1,5),(1,6),(2,4),(2,5),(2,6),(3,4),(3,5),(3,6)\}$ and $I_2 = \{(1,2),(1,3),(2,3),(4,5),(4,6),(5,6)\}$. The profile of feature j can be modeled by a set of binary values. More precisely, $P_j = \{a_{ij} \in \{0,1\}, \forall i \in I_1 \cup I_2, \forall j \in J\}$. Therefore, for element $i \in I_1$ (pairs of entities of different groups), if feature j has different values of expression level for the pair (for example, one entity has a value of 1 and the other 0), then $a_{ij} = 1$. Otherwise, $a_{ij} = 0$.

Following the problem's statement that a minimum cardinality set of features must be selected such that the similarities between entities of the same group and the differences between entities of different groups are maximized. We follow a three-phase decomposition-based approach for solving the (α,β)-k FSP, which has also been used in the study by Paula (2012). This three-phase approach decomposes the (α,β)-k FSP into three combinatorial optimization problems:

- Phase 1. Obtaining the maximum value of α (i.e. $\alpha^* \in \mathbb{Z}^+$) such that there exists a feasible solution for an instance of the (α,β)-k FSP. The value of α^* depends on the instance, however, α^* can be derived in polynomial time: $\alpha^* = \min_{i \in I_1}(\alpha_i)$, where $\alpha_i = \Sigma_{j \in J} a_{ij}, \forall i \in I_1$.

- Phase 2. Obtaining the minimum number of features k^* necessary to explain the dichotomy between the groups, considering that at least α^* features do so for each pair of entities of different groups. This problem is known the Min k (α,β)-k Feature Set Problem (FSP) (Paula, 2012). Any positive integer value less than α^* is still possible and will lead to a different value for k^*.

- Phase 3. Obtaining the maximum value of β (i.e. $\beta^* \in \mathbb{Z}^+$) such that a set of k^* features are selected to explain the dichotomy between the groups, and at least α^* features do so for each pair of entities of different groups. This problem is known the Max β (α,β)-k Feature Set Problem (FSP). This phase maximizes the internal consistency of the entities in the same group (a more robust feature set). Here, α^* and k^* are parameters.

3 Mathematical properties

We developed several properties for the Min k (α,β)-k FSP and Max β (α,β)-k FSP. Later in Section 4 we will utilize these properties in order to solve those problems.

Proposition 1. *An alternative optimal solution for the Min k (α,β)-k FSP can be obtained by including Equation* (3) *in integer program of the problem, and performing a re-optimization, where k^* is the optimal objective function value, and J^* is the set of features in an existing optimal solution.*

Proof. Observe that adding Equation (3) to the integer program of Min k (α,β)-k FSP ensures that the so obtained optimal solutions will not be explored.

$$\sum_{j \in J^*} x_j \neq k^* \tag{3}$$

After performing a re-optimization two outcomes are possible: (1) the re-optimization process leads to a new optimal solution for the Min k (α, β)-k FSP, which implies that we may keep obtaining a pool of optimal solutions (one new optimal solution per each re-optimization) or (2) the re-optimization process leads to an infeasible status, which means we have obtained all optimal solutions. ■

Proposition 2. *The Max β (α, β)-k FSP is the problem of selecting the best solution, among all optimal solutions of the Min k (α, β)-k FSP, according to the objective function of $z = \max \beta$.*

Proof. The proof is followed by observing that any optimal solution for the Min k (α, β)-k FSP is also feasible for the Max β (α, β)-k FSP. This is observed by the proposed three-phase solution method. ■

Proposition 3. *Given an optimal solution for the Min k (α, β)-k FSP, a feasible solution for the Max β (α, β)-k FSP may be obtained in polynomial time.*

Proof. Given an optimal solution for the Min k (α, β)-k FSP, we know that this solution is feasible for the Max β (α, β)-k FSP. Then, we need to calculate the value of β. Equation (4) shows this.

$$\beta = \min_{i \in I_2} \left(\sum_{j \in J^*} a_{ij} x_j \right) \tag{4}$$

Finally, notice that Equation (4) may easily be calculated in $O(n)$. ■

Proposition 4. *Given all optimal solutions for the Min k (α, β)-k FSP, the Max β (α, β)-k FSP will reduce to a sorting problem, and hence, can be solved in polynomial time.*

Proof. Proposition 2 states that the Max β (α, β)-k FSP is to select the best solution, according to maximizing the value of β, among all optimal solutions of the Min k (α, β)-k FSP. Given all optimal solutions of the Min k (α, β)-k FSP, we have a pool of all feasible solutions for the Max β (α, β)-k FSP, from which the solution with the maximum value of β can be performed by a sorting algorithm. ■

It should be noted that because the Min k (α, β)-k FSP is an integer program, obtaining all optimal solutions is generally $\mathcal{NP}$-Complete. Nevertheless, we discuss in Section 5 that even obtaining a few optimal solutions may lead to very good quality solutions for the Max β (α, β)-k FSP.

4 Proposed solution methods

This section develops heuristic algorithms to solve these problems.

4.1 Solving the Min k (α, β)-k Feature Set Problem

Algorithm 1 summarizes a heuristic algorithm for the Min k (α, β)-k FSP. The algorithm obtains a lower bound, and constructs a feasible solution by repairing the lower bound. The feasible solution is improved by applying a local search algorithm.

Algorithm 1: The heuristic algorithm for solving the Min k (α, β)-k FSP.

Input: An integer program of the problem; set J of features, and set I_1 of elements; parameter α.
Output: A feasible solution (set J^*) for the Min k (α, β)-k FSP.

Step 1: Solve linear programming relaxation, and let $\underline{\mathbf{x}}^* = \{x_j, \forall j \in J\}$ be the optimal solution;
if $x_j \in \{0, 1\}, \forall j \in J$ **then**
    Stop, an optimal solution is obtained;
end
else
    Step 2: Set certain x_j to take a value of 1, and solve a sub-problem over the set of available features and "yet to be covered" elements;
    if *the solution is not optimal* **then**
        Step 3: Apply the removal local search algorithm to remove redundant features;
    end
end

4.1.1 Obtaining a lower bound

In order to obtain a lower bound for the Min k (α, β)-k FSP we solve the linear programming relaxation of the Min k (α, β)-k FSP. Because the Min k (α, β)-k FSP is an unweighted problem where all features' costs are one, then a tighter integer lower bound may be obtained by $\lceil \underline{z}^* \rceil \in \mathbb{Z}^+$. Notice that solving the linear programming relaxation may not lead to an integer solution. If $0 < x_j < 1, \exists j \in J$, we have at least one non-negative variable. Therefore, this solution is not feasible for the Min k (α, β)-k FSP. Nevertheless, this procedure may result in a partially built solution for the Min k (α, β)-k FSP.

4.1.2 Obtaining a feasible solution

We build a feasible solution by keeping certain features into the solution, and solving a sub-problem (of the original problem) over available features and "yet to be covered" elements. Notice that because we keep a set of features in the solution, and that we do not have a guarantee that this set is part of an optimal set, redundant features may be forced into the solution. Hence, we may further improve the solution by removing redundant features. This is discussed in Section 4.1.3.

4.1.3 Improving the feasible solution

This procedure includes looking for redundant features in a feasible solution, and removing them in order to further improve the solution by investigating whether removing feature j from the feasible solution leaves the solution feasible. The stopping criterion of the algorithm is whenever removing features does not lead to a feasible solution or as soon as there is no redundant feature.

4.2 Solving the Max β (α, β)-k Feature Set Problem

We utilize Proposition 2, which states that the optimal solution for the Max β (α, β)-k FSP lies in the pool of all optimal solutions of the Min k (α, β)-k FSP, in order to design and implement an efficient heuristic algorithm for the Max β (α, β)-k FSP. The proposed heuristic algorithm combines both exact and heuristics, and has two major steps. Step 1 heuristically generates a feasible initial solution, and

Step 2 improves the solution by an exact method. Algorithm 2 summarizes the proposed procedure.

Algorithm 2: The heuristic algorithm to solve the Max β (α, β)-k FSP.

Input: Integer programs for the problems; set J of features; set $J^* = \{\}, J^* \subseteq J$ of selected features; sets I_1 and I_2 of elements; parameters α and p.

Output: An improved solution (set $J^* \subseteq J$ of features) for the Max β (α, β)-k FSP.

Step 1. Constructing a feasible initial solution.

Obtain a pool $P = \{P_1, \ldots, P_p\}$ of optimal solutions for the Min k (α, β)-k FSP;

Given $\tilde{J} \subset J$ the set of common features across all solutions of the pool, construct a partially built solution for the Max β (α, β)-k FSP by including $\tilde{J}$, i.e. $J^* = J^* \cup \tilde{J}$;

if J^* *is not feasible* **then**

  Solve a sub-problem (of the original problem) over the sets of available features and yet to be covered elements; let $\tilde{J}$ be the set of features in the optimal solution of the sub-problem;

  $J^* = J^* \cup \tilde{J}$;

end

Step 2. Improving the feasible initial solution.

Apply an exact solver/algorithm to solve the original Max β (α, β)-k FSP, given the set J^* of features as a starting solution;

Report J^*;

4.2.1 Constructing a feasible initial solution

From the pool of p optimal solutions the algorithm extracts those features that are common across all optimal solutions. Because these features may have a high probability to be in an optimal solution for the Max β (α, β)-k FSP, or at least it can be argued that they are part of a very good quality feasible solution. Let $\tilde{J} \subset J$ denotes the set of common features. Notice that $\tilde{J}$ may not be a feasible solution. Therefore, we need to add additional features in order to obtain a feasible solution.

Adding additional features may be performed by solving a sub-problem of the original Max β (α, β)-k FSP, which has a reduced number of features and elements because a set of features have already been chosen to be in a solution. Indeed, the sub-problem is generated by including sets of available features and yet to be covered elements. The union of the set of features obtained through solving this sub-problem and $\tilde{J}$ forms a feasible (initial) set of features for the Max β (α, β)-k FSP. If the sub-problem of Step 1 is large, and therefore cannot be solved in a short time, recursive applications of Algorithm 2 can be performed. If we cannot solve the Min k (α, β)-k FSP to optimality, we can utilize the best obtained solutions.

4.2.2 Improving the feasible initial solution

After generating a feasible solution for the Max β (α, β)-k FSP we can further improve the solution. Because our focus has been on benefiting the available standard solvers for this purpose, we provide the feasible initial solution as a starting solution to the solver CPLEX. In other words, Step 2 solves the original Max β (α, β)-k FSP, and may yield proven optimal solutions. It is worth emphasizing that according to our computational results of Section 5, the solver CPLEX is unable to solve large instances of Max β (α, β)-k FSP in a reasonable amount of time; even worse, it is unable to obtain feasible solutions for several instances within 10 hours of running. Having said that, Step 1 leads to very high quality solutions, particularly for large instances of Max β (α, β)-k FSP.

Table 2: Basic information of 11 real-world datasets.

Instance	No. of features	No. of entities	$\|J\|$	$\|I_1\|$	$\|I_2\|$	α^*	Reference
ADMF	686	83	686	1720	1683	86	Paula et al. (2011)
DS	73	15	73	56	49	50	Lockstone et al. (2007)
PD1	17099	105	17097	2750	2710	3970	Scherzer et al. (2007)
PD2	1674	25	1674	144	156	760	Lesnick et al. (2007)
PC	3556	171	3556	7290	7245	229	Chandran et al. (2007)
SM	525	1219	525	273834	468537	22	Charlesworth et al. (2010)
Class 0-vs-all	1969	450	1969	32400	68625	354	Haque et al. (2016)
Class 1-vs-all	3304	450	3304	32400	68625	683	Haque et al. (2016)
Class 2-vs-all	4243	450	4243	32400	68625	1016	Haque et al. (2016)
Class 3-vs-all	5436	450	5436	32400	68625	1394	Haque et al. (2016)
Class 4-vs-all	2005	450	2005	32400	68625	387	Haque et al. (2016)

5 Computational results

We implemented Algorithm 1 and Algorithm 2 in the programming language Python 2.7 via the solver CPLEX 12.5.0 Python API. The computing resource has Linux Ubuntu 14.04 LTS operating system with 32 GB of memory and 12 cores of Intel®Xeon CPU E5-1650 at 3.5 GHz; however, for all computational experiments we utilized only one thread (processor). Two sets of real-world instances were considered to evaluate the performance of the algorithms. The first set, which includes six biological instances ranging from small to large, was previously studied by Paula (2012), and the second set, which includes five large face recognition instances, truly represents actual size of the datasets we may encounter in applications of the (α, β)-k FSP. Obtaining optimal solution for the instances of the second set, or even good quality solutions, has been a challenge for the CPLEX.

Basic information regarding these instances is shown in Table 2. The first three columns show the instance name, number of features, and total number of entities/samples (of both group 1 and group 2). In each dataset, we have two groups of data: group 1 (e.g. Healthy or Control) and group 2 (e.g. Disease or Case; see Section 2 for more details). The next four columns provide parameters of the Min k (α, β)-k FSP and Max β (α, β)-k FSP associated with each instance. Here, column "$|J|$" gives the total number of features, column "$|I_1|$" and "$|I_2|$" are total number of pairs of entities of different and of the same groups. Column "α^*" shows the optimal value of parameter α.

5.1 Results of solving the Min k (α, β)-k Feature Set Problem

Table 3 reports the computational results of Algorithm 1 on those 11 instances. The heuristic algorithm is very competitive compared to the standard solver CPLEX, and obtains very high quality solutions. Several points are worth discussing. First, performance of the heuristic algorithm is very close to the solver CPLEX, whereas it requires almost 14 times less computation time. Second, on average, Algorithm 1 obtains solutions within 0.02% from optimality, which is quite promising. Third, the number of non-integer variables (shown in column "No. of non-integers") is a tiny fraction of the total number of variables, in particular, for large instances. This shows the role of the lower bound in constructing a partially built solution. Fourth, the reported lower bound (shown in column "LB") is of excellent quality: except for instances SM and Class 4-vs-all, for the remaining instances the value of lower bound equals the optimal solution. More importantly, for these two instances the lower bound is very close to the optimal solution.

Columns "CPLEX" refer to the outcomes of the solver CPLEX including the objective function

Table 3: Computational results for solving 11 instances of Min k (α, β)-k FSP, where $\alpha = \alpha^*$.

Instance	α^*	k^*	CPLEX			Heuristic (Algorithm 1)				
			k	Time(s)	Gap(%)	k	Time(s)	Gap(%)	No. of non-integers	LB
ADMF	86	292	292	0.57	0	292*	1.29	0.00	21	292
DS	50	65	65	0	0	65*	0.03	0.00	0	65
PD1	3970	9807	9807	106.59	0	9808	80.36	0.01	39	9807
PD2	760	1265	1265	0.11	0	1265*	0.88	0.00	0	1265
PC	229	725	725	118.41	0	726	21.94	0.14	39	725
SM	22	128	128	593.12	0	128*	129.87	0.00	42	127
Class 0-vs-all	354	1116	1117	36000	0.11	1116*	155.96	0.09	53	1116
Class 1-vs-all	683	2220	2220	375.16	0	2220*	284.75	0.00	38	2220
Class 2-vs-all	1016	3154	3154	1998	0	3155	651.47	0.03	38	3154
Class 3-vs-all	1394	4395	4395	3305.22	0	4395*	1368.77	0.00	32	4395
Class 4-vs-all	387	1324	1324	208.14	0	1324*	150.33	0.00	33	1323
Average				3882.30	0.01		258.70	0.02		

value, computation time in second, and optimality gap in %. For Algorithm 1, column "k" is the best objective function value obtained by the algorithm (the optimal solutions were highlighted by an "$*$"), "Time(s)" denotes the computation time in second, and "Gap(%)" is calculated as $\frac{k-k^*}{k^*} \times 100$, where k^* is the best available solution for the Min k (α, β)-k FSP. Also, "No. of non-integers" is the number of fractional variables (of solving the linear programming relaxation), and "LB" is the lower bound.

5.2 Results of solving the Max β (α, β)-k Feature Set Problem

Table 4 shows the outcomes of the proposed heuristic presented in Algorithm 2 over 11 instances of Table 2. According to Table 4, the heuristic algorithm obtains proven optimal solutions for all instances in the first set, as well for all instances in the second set except Class 0-vs-all. In addition to this, for all instances of the second set including Class 0-vs-all, the algorithm obtains better solutions than the CPLEX, and that in a much shorter time. We should emphasize that the instances of the second set are those that the solver CPLEX encounters a great difficulty in solving them.

We also reported the outcomes of the solver CPLEX (here, "-" denotes the CPLEX was stopped at the time limit of 36,000 seconds without obtaining even a feasible solution). Column $|\tilde{J}|$ shows the number of common features across all solutions of the pool, β_0 is the initial value of β, which is calculated for the feasible solution obtained in Step 1 of Algorithm 2, and β is the best value of the objective function, which is obtained in Step 2 (optimal values were highlighted by an asterisk). Columns "Time(s)" and "Gap(%)" denote the computation time in second, and gap calculated as $\frac{\beta-\beta^*}{\beta^*} \times 100$, where β^* is best values of β.

Note that the proposed heuristic algorithm obtains proven optimal solutions for 10 instances out of 11, whereas the solver CPLEX obtains optimal solution for only 6 instances. Furthermore, for instances Class 0-vs-all and Class 2-vs-all, the CPLEX is unable to deliver feasible solutions within 10 hours of running. Also, the average computation time of the heuristic algorithm is less than the CPLEX, while it solves large instances more quickly. Finally, observe that while the average optimality gap of CPLEX is 0.8% (calculated over nine solved instances), that of the heuristic algorithm is 0.06%.

Table 4: Computational results of solving 11 instances, where $\alpha = \alpha^*$, and $p = 20$ (20 optimal solutions for each instance of the Min k (α, β)-k FSP were obtained).

Instance	α^*	β^*	CPLEX			Heuristic (Algorithm 2)				
			β	Time(s)	Gap(%)	$\lvert\tilde{J}\rvert$	β_0	β	Time(s)	Gap(%)
ADMF	86	118	118	5.84	0.00	101	114	118*	13.11	0.00
DS	50	51	51	0.02	0.00	51	51	51*	0.16	0.00
PD1	3970	4325	4325	2013.78	0.00	8858	4324	4325*	3489.94	0.00
PD2	760	645	645	0.4	0.00	1265	645	645*	14.56	0.00
PC	229	233	233	18539.13	0.00	225	233	233*	3657.76	0.00
SM	22	40	40	10577.19	0.00	37	39	40*	6579.08	0.00
Class 0-vs-all	354	471	-	36000	-	998	471	471	2466.08	0.63
Class 1-vs-all	683	989	987	36000	0.41	2120	982	989*	2428.11	0.00
Class 2-vs-all	1016	1394	-	36000	-	3005	1394	1394*	3801.78	0.00
Class 3-vs-all	1394	1965	1964	11044.41	0.33	4215	1962	1965*	7642.15	0.00
Class 4-vs-all	387	549	549	1877.36	0.00	501	536	549*	3935.98	0.00
Average				13823.47	0.08				3093.52	0.06

6 Conclusion

This study contributed into the solution methods for the (α, β)-k Feature Set Problem (FSP). In order to solve the (α, β)-k FSP, we followed an existing three-phase approach. We solved the Phase 1 in polynomial time, and developed two heuristic algorithms for Phases 2 and 3. The proposed heuristic algorithms were tested over a set of 11 real-world instances. We showed that the outcomes of algorithms are superior than those of solver CPLEX, in terms of both solution quality and computation time, and additionally they include solving several instances that the CPLEX has shown to be unable to deliver feasible solutions within 10 hours of running.

Acknowledgments

Dr Amir Salehipour is the recipient of an Australian Research Council Discovery Early Career Researcher Award (DECRA) funded by the Australian Government.

References

Albrecht, A. A. (2006). "Stochastic local search for the feature set problem, with applications to microarray data". In: *Applied Mathematics and Computation* 183(2), pp. 1148–1164.

Berretta, R., Costa, W., and Moscato, P. (2008). "Combinatorial Optimization Models for Finding Genetic Signatures from Gene Expression Datasets". In: *Bioinformatics: Structure, Function and Applications. Series: Methods in Molecular Biology* 453(01), pp. 363–377.

Chandran, U. R., Ma, C., Dhir, R., Bisceglia, M., Lyons-Weiler, M., Liang, W., Michalopoulos, G., Becich, M., and Monzon, F. A. (2007). "Gene expression profiles of prostate cancer reveal involvement of multiple molecular pathways in the metastatic process". In: *BMC cancer* 7(1), p. 64.

Charlesworth, J. C., Curran, J. E., Johnson, M. P., Göring, H. H., Dyer, T. D., Diego, V. P., Kent, J. W., Mahaney, M. C., Almasy, L., MacCluer, J. W., et al. (2010). "Transcriptomic epidemiology of smoking: the effect of smoking on gene expression in lymphocytes". In: *BMC medical genomics* 3(1), p. 29.

Cotta, C, Sloper, C, and Moscato, P (2004). “Evolutionary search of thresholds for robust feature set selection: Application to the analysis of microarray data”. In: *Applications of Evolutionary Computing*. Ed. by G. Raidl. Vol. 3005. Lecture Notes in Computer Sscience. EvoWorkshops Conference, Coimbra, Portugal, April 05-07, 2004, pp. 21–30.

Fan, Y.-J. and Chaovalitwongse, W. A. (2010). “Optimizing feature selection to improve medical diagnosis”. In: *Annals of Operations Research* 174(1), pp. 169–183.

Ferchichi, S. E., Laabidi, K., and Zidi, S. (2009). “Genetic Algorithm and Tabu Search for Feature Selection”. In: *Studies in Informatics and Control* 18(2), pp. 181–187.

Haque, M. N., Noman, N., Berretta, R., and Moscato, P. (2016). “Heterogeneous Ensemble Combination Search Using Genetic Algorithm for Class Imbalanced Data Classification”. In: *PLoS ONE* 11(1), e0146116.

Inostroza-Ponta, M., Electrical Engineering, U. of Newcastle (N.S.W.). School of, and Science, C. (2008). “An integrated and scalable approach based on combinatorial optimization techniques for the analysis of microarray data”. English. School of Electrical Engineering and Computer Science. Thesis.

Inostroza-Ponta, M., Berretta, R., and Moscato, P. (Jan. 2011). “QAPgrid: A Two Level QAP-Based Approach for Large-Scale Data Analysis and Visualization”. In: *PLOS ONE* 6(1), pp. 1–18.

Lesnick, T. G., Papapetropoulos, S., Mash, D. C., Ffrench-Mullen, J., Shehadeh, L., Andrade, M. de, Henley, J. R., Rocca, W. A., Ahlskog, J. E., and Maraganore, D. M. (June 2007). “A Genomic Pathway Approach to a Complex Disease: Axon Guidance and Parkinson Disease”. In: *PLoS Genet* 3(6), e98.

Lockstone, H., Harris, L., Swatton, J., Wayland, M., Holland, A., and Bahn, S. (2007). “Gene expression profiling in the adult Down syndrome brain”. In: *Genomics* 90(6), pp. 647–660.

Meiri, R and Zahavi, J (2006). “Using simulated annealing to optimize the feature selection problem in marketing applications”. In: *European Journal of Operational Reserach* 171(3). 21st Euro Summer Institute (ESI), Nida Neringa, LITHUANIA, JUL 25-AUG 07, 2003, pp. 842–858.

Paula, M. Rocha de (2012). “Efficient Methods of Feature Selection Based on Combinatorial Optimization Motivated by the Analysis of Large Biological Datasets”. PhD thesis. School of Electrical Engineering and Computer Science, The University of Newcastle, Australia.

Paula, M. Rocha de, Gómez Ravetti, M., Berretta, R., and Moscato, P. (Mar. 2011). “Differences in Abundances of Cell-Signalling Proteins in Blood Reveal Novel Biomarkers for Early Detection Of Clinical Alzheimer’s Disease”. In: *PLoS ONE* 6(3), e17481.

Ravetti, M. G. and Moscato, P. (2008). “Identification of a 5-protein biomarker molecular signature for predicting Alzheimer’s disease”. In: *PLoS One* 3(9), e3111.

Ravetti, M. G., Berretta, R, and Moscato, P (2009). “Novel Biomarkers for Prostate Cancer Revealed by (α,β)-k-Feature Sets”. In: vol. 5. Foundations of Computational Intelligence. Springer Berlin / Heidelberg. Chap. 7, in Foundations of Computational Intelligence, pp. 149–175.

Ravetti, M. G., Rosso, O. A., Berretta, R., and Moscato, P. (Apr. 2010). “Uncovering Molecular Biomarkers That Correlate Cognitive Decline with the Changes of Hippocampus’ Gene Expression Profiles in Alzheimer’s Disease”. In: *PLoS ONE* 5(4), e10153.

Scherzer, C. R., Eklund, A. C., Morse, L. J., Liao, Z., Locascio, J. J., Fefer, D., Schwarzschild, M. A., Schlossmacher, M. G., Hauser, M. A., Vance, J. M., Sudarsky, L. R., Standaert, D. G., Growdon, J. H., Jensen, R. V., and Gullans, S. R. (2007). “Molecular markers of early Parkinson’s disease based on gene expression in blood”. In: *Proceedings of the National Academy of Sciences* 104(3), pp. 955–960.

Tsai, C.-F. and Hsiao, Y.-C. (2010). “Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches”. In: *Decision Support Systems* 50(1), pp. 258–269.