OPEN ACCESS Freely available online

PLoS one

Hierarchical Clustering Using the Arithmetic-Harmonic Cut: Complexity and Experiments

Romeo Rizzi[1], Pritha Mahata[2], Luke Mathieson[3,4]*, Pablo Moscato[3,4]

1 Dipartimento di Matematica ed Informatica, University of Udine, Udine, Italy, **2** New South Wales Rural Doctors Network, Newcastle, Australia, **3** Centre for Bioinformatics, Biomarker Discovery & Information Based Medicine, The University of Newcastle, Callaghan, Australia, **4** Information Based Medicine Program, Hunter Medical Research Institute, Newcastle, Australia

Abstract

Clustering, particularly hierarchical clustering, is an important method for understanding and analysing data across a wide variety of knowledge domains with notable utility in systems where the data can be classified in an evolutionary context. This paper introduces a new hierarchical clustering problem defined by a novel objective function we call the *arithmetic-harmonic cut*. We show that the problem of finding such a cut is NP-hard and APX-hard but is fixed-parameter tractable, which indicates that although the problem is unlikely to have a polynomial time algorithm (even for approximation), exact parameterized and local search based techniques may produce workable algorithms. To this end, we implement a memetic algorithm for the problem and demonstrate the effectiveness of the arithmetic-harmonic cut on a number of datasets including a cancer type dataset and a corona virus dataset. We show favorable performance compared to currently used hierarchical clustering techniques such as k-Means, Graclus and Normalized-Cut. The arithmetic-harmonic cut metric overcoming difficulties other hierarchal methods have in representing both intercluster differences and intracluster similarities.

Citation: Rizzi R, Mahata P, Mathieson L, Moscato P (2010) Hierarchical Clustering Using the Arithmetic-Harmonic Cut: Complexity and Experiments. PLoS ONE 5(12): e14067. doi:10.1371/journal.pone.0014067

Editor: Vladimir Brusic, Dana-Farber Cancer Institute, United States of America

Received July 21, 2010; **Accepted** October 28, 2010; **Published** December 2, 2010

Funding: We acknowledge the support of the Australian Research Council (ARC) Centre of Excellence in Bioinformatics (CE0348221), The University of Newcastle, and ARC Discovery Project DP0773279 (Application of novel exact combinatorial optimisation techniques and metaheuristic methods for problems in cancer research). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: luke.mathieson@newcastle.edu.au

Introduction

The problem of finding structure in a set of unlabeled data (the so-called *clustering problem*) appears in various domains of research including bioinformatics, machine learning, image processing and video processing. In the area of bioinformatics, clustering has become increasingly important, as finding genetic subtypes of heterogeneous diseases like breast cancer, ovarian cancer and multiple sclerosis, may be made easier by using suitable clustering methods. This work aims to facilitate this line of research by finding good clusterings of various datasets with known partitions.

The importance of the clustering problem in various areas has given rise to several greedy algorithms such as k-Means, optimization-based methods such as Normalized-Cut and neural-net based methods amongst others.

In this work, we will use a top-down approach for hierarchical clustering, recursively dividing the elements in the data. In each division step, often a graph partitioning technique is used (a similar approach is used for Normalized-Cut [1]). However, many graph (bi)partitioning problems can be formulated as NP-hard optimization problems, for which there are no polynomial-time algorithms to find the optimal solution unless $P=NP$. This is an indication of the difficulty of the clustering problem and the focus of research since the work of Wertheimer [2]. In this work, we propose a new objective function for graph bipartitioning. The motivation for finding a new objective function for graph bipartitioning is that the known bipartitioning methods produce incorrect results for some datasets. For example, two formulations for clustering by graph partitioning are Max-Cut [3] and Normalized-Cut [1]. Max-Cut is already known to provide incorrect results for some datasets [1]. We show that Normalized-Cut also produces some incorrect results for some of the datasets examined.

To achieve a better clustering than agglomerative hierarchical clustering and existing graph partitioning formulations, our proposed objective function seeks to minimize the intra-cluster distances and at the same time it seeks to maximize the inter-cluster distances. Our objective function performs well in clustering diverse types of datasets.

More precisely, we pose the hierarchical clustering problem as a finite number of instances of a graph partitioning problem, called Arithmetic-Harmonic Cut (AH-Cut). In the AH-Cut problem, we start with a distance matrix for a set of objects and compute a weighted graph in which vertices represent objects and edges are weighted by the distance between the corresponding vertices. Our objective function tries to obtain a partition where the weight of the partition is directly proportional to the sum of the weights on the edges between the two partite sets and the sum of the reciprocals of the weights on the edges inside the partite sets. When considered as an optimisation problem, the goal is to maximise the weight of the partition. The recursive application of AH-Cut can then be used to generate a tree-based classification of the data.

As noted many graph bipartioniting problems are NP-hard at least, so a theoretical examination of any proposed clustering

problem is necessary to determine whether it constitutes a practical approach to clustering. We give such a classification of AH-Cut and show that although it is *NP*-hard and hard to approximate, it is *fixed-parameter tractable*, and therefore still a practical method for clustering.

Related Objective Functions for Hierarchical Clustering

The k-Means Algorithm. The k-Means algorithm is one of a group of algorithms called *partitioning methods*; Given n objects in a d-dimensional metric space, we wish to find a partition of the objects into k groups, or clusters, such that the objects in a cluster are more similar to each other than to objects in different clusters. The value of k may or may not be specified and a clustering criterion, typically the squared-error criterion, must be adopted. The k-Means algorithm initializes k clusters by arbitrarily selecting one object to represent each cluster. Each of the remaining objects are assigned to a cluster and the clustering criterion is used to calculate the cluster mean. These means are used as the new cluster points and each object is reassigned to the cluster that it is most similar to. This continues until there is no longer a change when the clusters are recalculated. However, it is well-known that depending on the initial centres of the clusters, clustering results can change significantly. We use Gene Cluster 3.0 [4] for comparing our method with k-Means.

Max Cut, Ratio Cut and Average Cut. Graph bipartitioning algorithms are also used for clustering [5]. Given a graph $G=(V,E)$ and perhaps a weighting function $\omega : E \rightarrow S \subseteq \mathbb{R}$, a graph bipartitioning problem asks for a partition $S \uplus S' = V$ such that some function on the (weights of the) edges between S and S' satisfies the given bound, or in the case of an optimisation problem, is optimised. One of the most common formulations is essentially an *NP*-hard combinatorial problem, called Weighted Max-Cut, which is a simple weighted extension of the Max-Cut problem. If we denote the edges between S and S' as $E_{SS'}$ then the function $f_m(S,S')$ to be optimised in the case of Weighted Max-Cut is:

$$f_{mc}(S,S') = \sum_{e \in E_{SS'}} \omega(e).$$

Although good algorithms exist for Weighted Max-Cut, Shi and Malik [1] and Wu and Leahy [5] show that (Weighted) Max-Cut 's objective function favours cutting small sets of isolated nodes in the graph. Furthermore, during bipartitioning, sometimes it may also cut small groups and put two parts of the same small group into different partite sets.

Ratio-Cut uses the objective function:

$$f_{rc}(S,S') = \frac{\sum_{e \in E_{SS'}} \omega(e)}{\min\{|S|,|S'|\}}.$$

In this case ω is taken as a similarity metric. Ratio-Cut (and its k-way extension) has also been employed for image segmentation [6] and circuit partitioning for hierarchical VLSI design [7].

Average-Cut employs the following objective funtion:

$$f_{ac}(S,S') = \left(\sum_{e \in E_{SS'}} \omega(e) \right) \left(\frac{1}{|S|} + \frac{1}{|S'|} \right).$$

If ω is a similarity metric, the the problem becomes a minimisation problem, ω expresses distance the goal is maximisation. It turns out that even using the average cut, one cannot simultaneously minimise the inter-cluster similarity while maximizing the similarity within the groups.

Normalized-Cut. In the context of image segmentation, Shi and Malik [1] introduce Normalized-Cut. They use a similarity metric for ω, and thus Normalized-Cut is typically expressed as a minimisation problem with the following objective function:

$$f_{nc}(S,S') = \left(\sum_{e \in E_{SS'}} \omega(e) \right) \left(\frac{1}{\sum_{i \in S, j \in V} \omega(ij)} + \frac{1}{\sum_{i \in S', j \in V} \omega(ij)} \right).$$

It is well-known that by negating weights the Max-Cut problem is equivalent to the corresponding Min-Cut problem where one is supposed to minimise the sum of the weights (given by some similarity measure) between the two partitions $(S,\overline{S})$ of a set of vertices V in a graph $G=(V,E)$. It is straightforward to see that the same argument holds in case of Normalized-Cut as well, which allows the negation of a distance matrix to be used a similarity matrix, facilitating comparisons for datasets for which only distance matrices are available. However, Shi and Malik [1] start with a Euclidian distance matrix D and then compute e^{-D} as their similarity matrix. We use both approaches and demonstrate that the performance of this algorithm varies depending on the dataset and the two similarity measures.

Furthermore, Shi an Malik's [1] implementation relaxes the Normalized-Cut problem into a generalised eigen-value problem by allowing the vertices v to take real-values, instead of taking values from just the set $\{0,1\}$ where $v=0$ denotes that $v \in S$ and $v=1$ denotes that $v \in S'$. Then, for bipartitioning, the second smallest eigenvector of the generalized eigen system is the real-valued solution to Normalized-Cut. Finally, they search for the splitting point as follows: first choose l equidistance splitting points, compute Normalized-Cut's objective value for each of these splits, then choose the one for which Normalized-Cut's objective value is the smallest. In fact, the implementation also allows k-way Normalized-Cut, Yu and Shi [8] examine this further. It considers the first k eigen vectors and yields k partite sets from a discretisation step following it.

Notice that Max-Cut and Ratio-Cut do not cluster by intra-cluster similarity and this results in a poor clustering results for image segmentation in comparison to Normalized-Cut [1]. Therefore, among these three algorithms, we consider only Normalized-Cut for comparison with our algorithm.

Graclus. Graclus [9] implements a multilevel algorithm that directly optimizes various weighted graph clustering objectives, such as the popular ratio cut, normalized cut, etc. This algorithm for multilevel clustering consists of three steps: (a) iteratively merging nodes of the graph (using various criteria of merging) and creating supergraphs with fewer nodes; (b) applying some base-level clustering on the resulting supergraph; and (c) restoring the clusters of original graph iteratively from the clustering of the final supergraph. This algorithm does not use eigenvector computation, and is thus notably faster than existing implementations of normalised and ratio-cuts. However, in most of the examples shown in this paper, Shi and Malik's [1] implementation of Normalized-Cut results in a better clustering than Graclus.

Outline of the Paper

In this paper, after introducing the problem, we first examine the approximability of AH-Cut. In fact, we prove that AH-Cut is *APX*-hard (and *NP*-complete) via a reduction from the Max-Cut problem, which is already known to be *APX*-hard [3]. Therefore

AH-CUT has no polynomial time approximation scheme unless $P=NP$. We then demonstrate that AH-Cut is fixed-parameter tractable via a greedy localisation algorithm. Such a complexity analysis provides an indication of what practical methods are suitable for application to the problem. In this case the complexity results indicate that there is unlikely to be a polynomial time algorithm (or even approximation), but that the exponential component of the running time is at worst only a function of a small independent parameter and therefore the problem is likely to still have effective algorithms.

Given the complexity result we use a meta-heuristic approach (namely, a *memetic algorithm*) for AH-Cut (an outline of which was presented previously [10]). We compare the performance of our algorithm on four diverse types of datasets and compare the results with two recent and highly regarded clustering algorithms: Normalized-Cut; and k-Means. The results indicate that AH-Cut gives a robust and broadly useful hierarchical clustering method.

Preliminaries

Graph Notation and Problem Definition. We consider only simple, undirected graphs, which may or may not be associated with a weight function on the edges. Given a graph G unless otherwise specified we denote the vertex set of G by $V(G)$ and the edge set of G by $E(G)$. We denote an edge between vertices u and v by uv.

Given a graph G and two vertex sets X and Y we denote the set of edges with one endpoint in X and the other in Y by $E_{XY}(G)$. When the graph is clear from context we write E_{XY}.

Arithmetic-Harmonic Cut (AH-Cut)
Instance: A graph $G=(V,E)$, two positive integers k and d and a weight function $\omega : E\rightarrow[1,d]$.
Question: Is there a partition of V into two sets B and W such that

$$\left(\sum_{uv\in E_{BW}}\omega(uv)\right)\left(\sum_{uv\in E\setminus E_{BW}}\frac{1}{\omega(uv)}\right)\geq k?$$

Given a graph G and two disjoint vertex sets X and Y, for convenience we denote

$$\left(\sum_{uv\in E_{XY}}\omega(uv)\right)\left(\sum_{uv\in E\setminus E_{XY}}\frac{1}{\omega(uv)}\right)$$

by

$$f_G(X,Y).$$

The optimisation verion of AH-Cut is identical except that we maximise the function f.

Approximation and Complexity. If a maximisation problem Π with objective function f has an polynomial time algorithm which given an instance I with optimal solution S guarantees a solution S^* where $f(S^*)\leq(1-\varepsilon)f(S)$ for some $\varepsilon>0$ then we say Π has a *constant factor approximation algorithm*. If there is an algorithm that guarantees such a bound for every $\varepsilon>0$, Π has a *polynomial time approximation scheme* (*ptas*). APX is the class of all problems which have constant factor approximation algorithms. If a problem Π is APX-hard, then Π has no ptas unless $P=NP$.

We refer to Ausiello *et al.* [11] for further reading.

Parameterized Complexity. A parameterized problem is a (decision) problem equipped with an additional input called the *parameter*. Typically the parameter will numeric and should be independent of the size of the instance and relatively small. A problem Π is *fixed-parameter tractable* if there is an algorithm that solves the problem in time bounded by $f(k)p(n)$ where k is the parameter, n is the size of the input, f is a computable function and p is a polynomial.

As we do not require the parameterized notion of hardness, we refer the reader to Flum and Grohe [12] for complete coverage.

Results and Discussion

The Complexity of AH-Cut

We first turn to theoretical results for AH-Cut. We show that the optimisation version of the problem is APX-hard, and consquently that the decision version is NP-complete, indicating that AH-Cut is not has no polynomial time algorithm, but has no polynomial time approximation scheme, under standard complexity theoretic assumptions. Under the parameterized complexity framework however we show that AH-Cut is fixed parameter tractable with a $2^{O(dk)}+|V|^3$ time algorithm.

NP-Completeness and APX-Hardness

We demonstrate the NP-completeness for AH-Cut via an APX-hardness reduction from Max-Cut which is known to be APX-hard [3] and NP-complete [13].

Max-Cut
Instance: A graph $G=(V,E)$, a positive integer k.
Question: Is there a set $S\subseteq V$, where $S'=V\setminus S$ such that $|E_{SS'}|\geq k$?

The goal of the optimisation version of Max-Cut is to maximise $|E_{SS'}|$.

Let (G,k) be an instance of Max-Cut with $m=|E(G)|$ and $|V(G)|\leq m\leq|V(G)|^2$ (we may assume that there is at least one cycle, as the maximum cut of any forest is trivially $E(G)$), we construct an instance (G',k') of AH-Cut where $V(G')=V(G)\cup\{a,b,c\}$ and $E(G')=\{uv|u,v\in V(G')\}$ (i.e., G' is a complete graph). The elements of $E(G')$ are weighted as follows: if $uv\in E(G)$, then we set $\omega(uv):=m^6$, if $u,v\in\{a,b,c\}$ we set $\omega(uv):=1$ and for all other edges uv we set $\omega(uv):=m^3$. We set $k':=3km^6$. Clearly we can obtain G' in polynomial time.

Before moving to the hardness proof, we first prove some auxilliary lemmas.

Lemma 1. *Let $S\subseteq V(G)$ and $S'=V(G)\setminus S$ where $|E_{SS'}|\geq k$, then $f_{G'}(S,S'')\geq 3km^6$ where $S''=S'\cup\{a,b,c\}$.*

Proof. Consider the objective function

$$f_{G'}(S,S'')=\left(\sum_{uv\in E_{SS''}}\omega(uv)\right)\left(\sum_{uv\in E\setminus E_{SS''}}\frac{1}{\omega(uv)}\right)$$

and let $A=\left(\sum_{uv\in E_{SS''}}\omega(uv)\right)$ and $B=\left(\sum_{uv\in E\setminus E_{SS''}}\frac{1}{\omega(uv)}\right)$.

Each of the edges between S and S'' that are also in $E(G)$ contribute m^6 to A, and all other edges that are also in $E(G)$ contribute $\frac{1}{m^6}$ to B. As a, b and c are in S'', the edges ab, ac and

bc contribute 3 to B. The edges between $\{a,b,c\}$ and S contribute $3m^3|S|$ to A. Therefore

$$f_{G'}(S,S'')=AB=\left(3m^3|S|+\sum_{uv\in E_{SS''}}m^6\right)\left(3+\sum_{uv\in E\setminus E_{SS''}}\frac{1}{m^6}\right).$$

As $|E_{SS''}|\geq|E_{SS'}|\geq k$,

$$f_{G'}(S,S'')\geq km^6\left(3+\sum_{uv\in E\setminus E_{SS''}}\frac{1}{m^6}\right)\geq 3km^6.$$

Lemma 2. *Assume $|E(G)|\geq 3$. Let T be a subset of $V(G')$ and $T'=V(G')\setminus T$. If $E_{TT'}\cap\{ab,ac,bc\}\neq\emptyset$ then $f_{G'}(T,T')<\frac{3}{2}m^7$.*

Proof. Assume $E_{TT'}\cap\{ab,ac,bc\}\neq\emptyset$, without loss of generality (by switching T and T') we may assume that $|T\cap\{a,b,c\}|=2$. Let $R=T\setminus\{a,b,c\}$, $R'=V(G)\setminus R$ and $t=|E_{RR'}(G)|$. Let $A=\left(\sum_{uv\in E_{TT'}}\omega(uv)\right)$ and $B=\left(\sum_{uv\in E\setminus E_{TT'}}\frac{1}{\omega(uv)}\right)$.

As $|T\cap\{a,b,c\}|=2$ we know $|E_{TT'}\cap\{ab,ac,bc\}|$, which contributes 2 to A. The edges in $E_{RR'}$ contribute tm^6 to A. As two vertices from $\{a,b,c\}$ are in T, the edges between those two vertices and R' contribute $2|R'|m^3=2(|V(G)|-|R|)m^3$ to A. The third vertex in $\{a,b,c\}$ contributes $|R|m^3$ to A.

One of the edges of $\{ab,ac,bc\}$ is not in $E_{TT'}$ and thus contributes 1 to B. There are $m-t$ edges in $E(G)$ that are not in $E_{RR'}$ (and thus not in $E_{TT'}$) and so contribute $\frac{m-t}{m^6}$ to B. The edges between the two $T\cap\{a,b,c\}$ vertices and R contribute $\frac{2|R|}{m^3}$ to B. Finally the edges between the $T'\cap\{a,b,c\}$ vertex and R' contribute $\frac{|V(G)|-|R|}{m^3}$ to B. Thus in total we have

$$A=(2+tm^6+2(|V(G)|-|R|)m^3+|R|m^3)$$

and

$$B=\left(1+\frac{m-t}{m^6}+\frac{2|R|}{m^3}+\frac{|V(G)|-|R|}{m^3}\right).$$

As $m\geq|V(G)|$,t and $|R|\leq|V(G)|$ we have

$$\begin{aligned}f_{G'}(T,T')&=AB\\&\leq(2+m^7+2m^4)\left(1+\frac{1}{m^5}+\frac{2}{m^2}\right)\\&\leq 2+m^7+2m^4+\frac{2}{m^5}+m^2+\frac{2}{m}+\frac{4}{m^2}+2m^5+4m^2\\&\leq 2+m^7+2m^4+1+m^2+1+1+2m^5+4m^2\\&\leq 5+m^7+2m^5+2m^4+5m^2\\&\leq m^7\left(\frac{5}{m^7}+\frac{2}{m^2}+\frac{2}{m^3}+\frac{5}{m^5}\right).\end{aligned}$$

As we assume that $m\geq 3$,

$$f_{G'}(T,T')\leq\frac{3}{2}m^7.$$

Lemma 3. *Assume $|V(G)|\geq 3$. Let S be a subset of $V(G')$ and $S''=V(G)\setminus S$ such that $f_{G'}(S,S'')\geq 3km^6$. In polynomial time we can obtain an $S^*\subset V(G)$ such that $|E_{SS*}|\geq k(1-\frac{5}{3m})$.*

Proof. If $k\leq\frac{m}{2}$ we may apply the greedy algorithm of Mahajan and Raman [14] which returns a set X such that $|E_{X(V(G)\setminus X)}|\geq\frac{m}{2}$. Therefore we may take X as S^* and we have $|E_{S^*(V(G)\setminus S^*)}|\geq\frac{m}{2}\geq k\geq k\left(1-\frac{5}{3m}\right)$.

If $k>\frac{m}{2}$, we have that $f_{G'}S,S''\geq\frac{3}{2}m^7$. Then by Lemma 2, $E_{SS''}\cap\{ab,ac,bc\}=\emptyset$. Without loss of generality we may assume that $S\cap\{a,b,c\}=\emptyset$ (by switching S and S'' as necessary). Denote $S''\setminus\{a,b,c\}$ by S'. As $m\geq|V|\geq 3$ we have $|V|^2\geq 3|V|$. We also have that $|V|\geq|S|$. We may then observe that

$$\begin{aligned}|E_{SS'}(G)|&=\sum_{uv\in E_{SS'}(G)}1\geq\sum_{uv\in E_{SS'}(G)}\left(1-\frac{1}{m}\right)\left(1+\frac{1}{m}\right)\\&\geq\left(1-\frac{1}{m}\right)\sum_{uv\in E_{SS'}(G')}\left(\frac{m^5}{m^6}+\frac{\omega(uv)}{m^6}\right)\\&\geq\left(1-\frac{1}{m}\right)\frac{1}{m^6}\left(|V|^2m^3+\sum_{uv\in E_{SS'}(G')}\omega(uv)\right)\\&\geq\left(1-\frac{1}{m}\right)\frac{1}{m^6}\left(3|S|m^3+\sum_{uv\in E_{SS'}(G')}\omega(uv)\right)\\&\geq\left(1-\frac{1}{m}\right)\frac{f_{G'}(S,S'')}{m^6\sum_{uv\in E(G')\setminus E_{SS'}(G')}\frac{1}{\omega(uv)}}.\end{aligned}$$

We know that $\frac{f_{G'}(S,S'')}{m^6}\geq 3k$, and that $\{ab,ac,bc\}$ contributes 3 edges to $E(G')\setminus E_{SS''}(G')$, as $k\geq\frac{m}{2}$ there are at most $\frac{m}{2}$ edges of G that are in $E_{SS'}(G')$ and there are at most $\binom{|V(G')|}{2}-\frac{m}{2}-3$ edges otherwise unnaccounted for in $E(G')\setminus E_{SS''}(G')$. Therefore:

$$|E_{SS'}(G)|\geq\left(1-\frac{1}{m}\right)\frac{3k}{3+\left(\binom{|V(G')|}{2}-\frac{m}{2}-3\right)\frac{1}{m^3}+\frac{m}{2m^6}}.$$

As $m\geq|V(G)|\geq 3$:

$$\begin{aligned}|E_{SS'}(G)|&\geq\left(1-\frac{1}{m}\right)\frac{3k}{3+\left(\binom{m+3}{2}\right)\frac{1}{m^3}+\frac{m}{2m^6}}\\&\geq\left(1-\frac{1}{m}\right)\frac{3k}{3+\frac{m^2+5m+6}{2m^3}+\frac{1}{2m^5}}\\&\geq\left(1-\frac{1}{m}\right)\frac{3k}{3+\frac{2}{m}}\\&\geq\left(1-\frac{1}{m}\right)\frac{3k}{3(1+\frac{2}{3m})}\\&\geq k\left(1-\frac{5}{3m}\right).\end{aligned}$$

We are now prepared to prove the main theorem of this section.

Theorem 4. AH-CUT *is APX-hard and NP-complete.*

Proof. Assume there is a $(1-\varepsilon)$-approximation algorithm $\mathcal{A}$ for AH-CUT. We show that this implies a $(1-2\varepsilon)$-approximation algorithm for MAX-CUT. Let (G,k) be an instance of MAX-CUT and (G',k') be the corresponding instance of AH-CUT derived from the reduction described above. If $|V(G)|\leq 3$ or $|E(G)|\leq \frac{5}{3\varepsilon}$, we solve the instance by complete enumeration in constant time. Otherwise assume the optimal cut of G cuts at least opt edges of $E(G)$ and induces the partition $V(G)=S\uplus S'$. By Lemma 1 the partition $V(G')=S\uplus V(G')\backslash S$ induces a solution for AH-CUT such that $f_{G'}(S,V(G')\backslash S)\geq 3|E(G)|^6\cdot opt$. Algorithm $\mathcal{A}$ will give a solution with $f_{G'}(S,V(G')\backslash S)\geq 3|E(G)|^6\cdot opt\cdot(1-\varepsilon)$. Then by Lemma 3 we have a set $S^*\subseteq V(G)$ such that

$$\begin{aligned}|E_{S^*(V(G)\backslash S^*}(G))| &\geq opt(1-\varepsilon)\left(1-\frac{5}{3m}\right)\\ &\geq opt(1-\varepsilon)(1-\varepsilon)\\ &\geq opt(1-2\varepsilon).\end{aligned}$$

It is clear that AH-CUT is in *NP*. Given a partition, we simply calculate f for that partition and compare to the target value.

Fixed-Parameter Tractability

We show that AH-CUT is fixed-parameter tractable via a greedy localisation technique. First we compute a greedy solution as follows:

1. Pick an edge $uv\in E$ such that $\omega(uv)\geq\omega(xy)$ for every $xy\in E$. Add u to B and v to W.
2. While $V\supset B\uplus W$ do
   (a) Pick a vertex $x\notin B\uplus W$ such that $x\in N(B\uplus W)$.
   (b) If $f(B\cup\{x\},W)\geq f(B,W\cup\{x\})$ then set $B:=B\cup\{x\}$, otherwise set $W:=W\cup\{x\}$.

Note that we assume that G is connected. If G is not connected then all vertices of degree 0 can be discarded, and the initial selection of vertices must take an adjacent pair from each connected component, then the algorithm continues as before. After all vertices have been assigned if $f(B,W)\geq k$, then we answer Yes. If $f(B,W)\leq k-1$ we make the following claim:

Lemma 5. *Let (G,k) be an instance of* AH-CUT *and $B\uplus W$ a partition of V such that $f(B,W)\leq k-1$, then $|V|\leq\max\{2dk,2(d+1)(k-1)\}$ and $|E|\leq\max\{dk,(d+1)(k-1)\}$.*

Proof. Let (G,k) be such an instance and $B\uplus W$ the partition.

If $\sum_{uv\in E\backslash E_{BW}}\frac{1}{\omega(uv)}<1$, then in particular we know that $\frac{1}{d}\sum_{uv\in E_{BW}}\omega(uv)\leq k-1$, therefore $\sum_{uv\in E_{BW}}\omega(uv)\leq d(k-1)$ and we have $|E_{BW}|\leq d(k-1)$. Furthermore if $\sum_{uv\in E\backslash E_{BW}}\frac{1}{\omega(uv)}<1$, then there are at most d edges in $E\backslash E_{BW}$. Thus the total number of edges is at most $d+d(k-1)$ and we have at most $2(d+d(k-1))$ vertices in the graph.

If $\sum_{uv\in E\backslash E_{BW}}\frac{1}{\omega(uv)}\geq 1$, then we immediately have that $\sum_{uv\in E_{BW}}\omega(uv)\leq k-1$ and therefore $|E_{BW}|\leq k-1$. The case with the most edges with both endpoints in the same partite set is then when there is only one edge between the two partite sets, therefore $\sum_{uv\in E\backslash E_{BW}}\frac{1}{\omega(uv)}\leq k-1$, then $|E\backslash E_{BW}|\leq d\cdot(\sum_{uv\in E\backslash E_{BW}}\frac{1}{\omega(uv)})$, therefore $|E\backslash E_{BW}|\leq d(k-1)$ and there are at most $k-1+d(k-1)$ edges and $2(k-1+d(k-1))$ vertices in the graph.

As the instance is bounded by a function of $k+d$, we can exhaustively search the instance in time $O(2^h)$ where $h=\max\{2dk,2(d+1)(k-1)\}$.

This algorithm immediately gives the following result:

Theorem 6. AH-CUT *is fixed-parameter tractable with an algorithm running in time $O(2^{\max\{2dk,2(d+1)(k-1)\}}n^3)$ where k is the optimisation target value, d is the maximum edge weight and n is the number of vertices in the input graph.*

AH-Cut in Practice

We apply our algorithm to four datasets: (i) melanoma-colon-leukemia data from National Cancer Institute, U.S [15] (involving gene expression of 6830 genes for 23 samples); (ii) SARS data of Yap *et al.* [16] and (iii) tissue type data given by Su *et al.* [17] (involving gene expression of 33689 genes for 158 tissue samples).

We also consider a large synthetic dataset consisting of 1000 samples and 500 features with a known optimal solution with three clusters. Despite the size of this datasets, our algorithm finds all three clusters.

In each case we compare our algorithm to NORMALIZED-CUT and where possible to k-MEANS and Graclus. Implementation details for the memetic algorithm are given in the Materials and Methods section.

Melanoma-Colon-Leukemia data from NCI

For the first comparison we use a subset of the data for 60 cancer samples taken for the National Cancer Institute's (NCI) screening for anti-cancer drugs [15]. The dataset consists of 6830 gene expressions of 8 melanoma, 7 colon tumour and 8 leukaemia samples. The reason for taking these three sets of samples is that others (non-small cell lung cancer, breast cancer, etc.) have heterogeneous profiles and removing these gives an expected solution of three clear clusters. Laan and Pollard [18] show that this simple dataset is already hard to cluster using agglomerative hierarchical clustering methods. Nevertheless, AH-CUT is able to cluster the samples of these three diseases effectively, see Figure 1 for the whole dendrogram generated by AH-CUT. We use centred correlation distance as the distance metric to maintain consistency with Golub *et al.* [19].

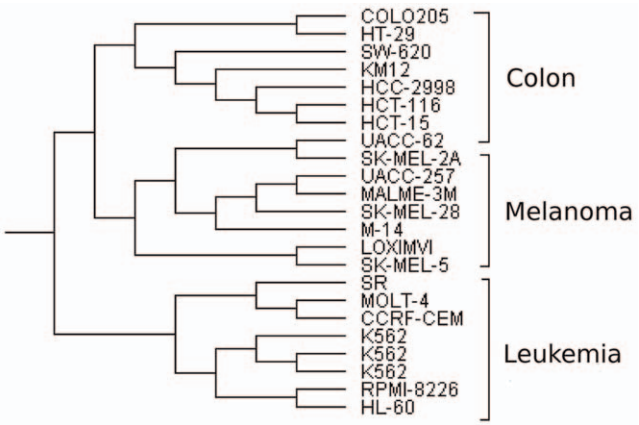

Figure 1. Dendrogram generated from AH-CUT for the melanoma-colon-leukemia dataset.
doi:10.1371/journal.pone.0014067.g001

Conversely, NORMALIZED-CUT behaves inconsistently in allocating samples to the partitions. Using the negated distance matrix as a similarity matrix and choosing two clusters, it either separates melanoma from colon and leukemia samples, or leukemia from colon and melanoma samples, or splits the leukemia sample group. Even when number of clusters is specified as 3, leukemia samples are split between different clusters. Using e^{-D} as the similarity matrix, where D is the distance matrix, gives worse results.

On the other hand, k-MEANS performs much better than NORMALIZED-CUT and successfully separates melanoma from colon and leukemia samples when $k=2$ and gives three distinct clusters of colon, melanoma and leukemia samples when $k=3$.

SARS

Next we analyse Yap *et al.*'s [16] dataset for Severe Acute Respiratory Syndrome (SARS). To explore the exact origin of SARS, the genomic sequence relationships of 31 different single-stranded RNA (ssRNA) viruses (both positive and negative strand ssRNA viruses) of various families were studied. Yap *et al.* [16] generate the tetra-nucleotide usage pattern profile for each virus from which a distance matrix based on correlation coefficients is created. We use this distance matrix for the following performance comparison of AH-CUT and NORMALIZED-CUT. See Figure 2 for the dendrogram generated by AH-CUT for this dataset. It is interesting to note that SARS virus is grouped in the same subtree as other corona viruses and is closest to the Feline Corona Virus (FCoV). Notice that these are all positive strand ssRNA viruses. This group of SARS and Corona viruses also contains other viruses (Porcine epidemic diarrhoea virus (PDV), Transmissible gastroenteritis virus (TGV), Avian infectious bronchitis virus (ABV), Murine hepatitis virus (MHV)). There is also a group of positive strand ssRNA viruses, called "Outliers", which exhibit differences in their tetra-nucleotide usage pattern from the rest. Yellow Fever Virus (YFV), Avian Encephalomyelitis Virus (AEV), Rabbit Hemorrhagic disease Virus (RHV), Equine arteritis Virus (EV1), Lactate Dehydrogenase-elevating Virus (LDV) were also identified as outliers by Yap *et al.* [16]. This group also includes other ssRNA positive strand viruses - Igbo ora virus (IOV), Bovine viral diarrheoa (BDV), Foot and mouth disease virus C (FMV) and Simina Haemorrhagic fever virus (SFV). The negative strand ssRNA viruses are clustered in two subgroups, one unmixed with the positive strand ssRNA viruses, the remainder in the group "Mixed". The first class (called -ve strand ssRNA viruses in Figure 2) covers Canine Distemper Virus (CDV) and Tioman virus (TV2), Reston Ebola Virus (REV), Bovine Ephemeral Fever Virus (BFV), Hantaan Virus (HV1), Bovine Respiratory syncytial Virus (BRV), Human Respiratory syncytial Virus (HRV) and Respiratory Syncytial Virus (RSV).

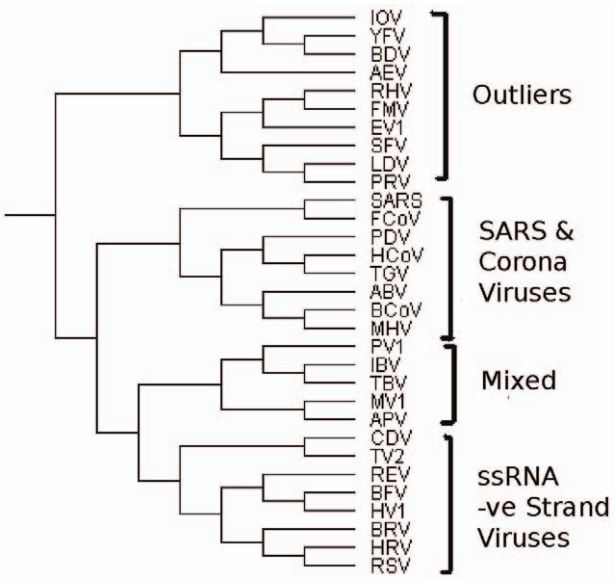

Figure 2. Dendrogram generated by AH-CUT for the SARS dataset.
doi:10.1371/journal.pone.0014067.g002

NORMALIZED-CUT mixes positive and negative strand ssRNA viruses even in the first partition for the majority of runs. Graclus puts corona viruses in different partitions even when the number of partition specified is 2. As Graclus requires a distance matrix of integers we scaled the dataset's distance matrix by 100 to obtain an integral distance matrix. As only a distance matrix was available, k-MEANS was not applicable.

Tissue dataset

Next we present results of applying AH-CUT to Su *et al.*'s [17] human tissue dataset. This dataset consists of 33689 tissue specific genes from 158 samples collected from 46 individuals. The known origin of tissue samples gives a great advantage in validating clusterings of this dataset. For brevity we will compare only the first partitioning of this dataset generated by the different algorithms. The first partition of AH-CUT consists of:

- brain related tissues;
- eye related tissues;
- face related tissues;
- testis tissues;
- others (including ovary tissue).

The second partition of AH-CUT consists of:

- bonemarrow related cells;
- blood cells;
- heart related cells;
- fœtal cells;
- others (including ovary tissue, uterine tissue and uterine corpus tissue).

The partitioning for this dataset is quite reasonable except occurrence ovary tissues in different partitions. This can be due to the possible outlier nature of ovary tissues. However, NORMALIZED-CUT repeatedly separates the brain related tissues and thus performs even worse. Graclus performs similarly to NORMALIZED-CUT on this dataset.

k-MEANS agrees very well with AH-CUT in the partitions, except that it clusters placental tissues within the second partition instead of the first and it puts the two uterine tissues in different partitions. k-MEANS also puts the two ovary tissues in two different partitions.

Synthetic large-sampled gene expression data

To test the scalability of our algorithm, we show the results of AH-CUT applied to a large synthetic dataset. Consider 1000 samples of 500 synthetic gene expression profiles corresponding to three subtypes of some disease, giving a known optimum clustering with three clusters. To generate the data, we follow Laan and Pollard's [18] method. We sample three groups of 700, 200 and 100 samples respectively from three multivariate normal distributions with diagonal covariance matrices, which differed only in their mean vector. The number of samples are chosen keeping in

mind that in general there are some predominant subtypes of a disease and some rarer subtypes. All genes had common standard deviation $\log_{10}(1.6)$, which corresponds to a 0.75-quantile of all standard deviations in an actual data set. For the first subpopulation, the first 25 genes had a mean of $\log_{10}(3)$, genes $25-50$ had mean of $-\log_{10}(3)$, and the other 350 genes had mean zero. Then for the second subpopulation, genes $51-75$ had mean of $\log_{10}(3)$, genes $76-100$ had mean of $-\log_{10}(3)$ and the other 350 genes had mean zero. For the third subpopulation, genes $101-125$ had mean of $\log_{10}(3)$, genes $126-150$ had mean of $-\log_{10}(3)$ and the other 350 genes had mean zero. In other words, signature of the three types of cancer is related to 50 genes of which 25 are under-expressed and 25 are over-expressed.

The application of AH-Cut on this dataset first separates the first group from the rest. A second application on the rest of the samples yields the second and third group as the two partitions.

When number of clusters is specified as two, Normalized-Cut clusters the first and third subtypes together and the second subtype separately. However, specifying number of clusters as three creates a partitioning which does not correspond to the expected known grouping.

k-Means puts the first subtype in one partition and the other two subtypes in another partition when $k=2$, and separates three subtypes successfully when $k=3$.

Conclusion

We have introduced a novel objective function for clustering based on graph partitioning. We show that the resulting problem AH-Cut is, unfortunately, *NP*-complete and *APX*-hard, but is however fixed-parameter tractable.

We then gave several test cases demonstrating the potential of the approach using a memetic algorithm. The performance of AH-Cut based clustering exceeds the performance of Normalized-Cut based clustering across a wide variety of datasets, including large scale datasets, and notably datasets with known optimal clusterings. AH-Cut based clustering also has a wider applicability than *k*-Means based clustering, and at least equal performance.

There are several avenues for further research from this initial exploration. The fixed-parameter tractability of AH-Cut promises the possibility of a practical *exact* algorithm, which would give stronger evidence of AH-Cut's performance, as random elements would be removed.

Further studies on datasets of all kinds would also be useful to explore the strengths and weaknesses of AH-Cut based clustering, especially in comparison to other existing methods.

Tangentially, the quality of the memetic algorithm solutions suggest that there may be a link between the fixed-parameter tractability and the performance of the memetic algorithm. As established by the fixed-parameter tractability of AH-Cut, if a simple greedy algorithm does not produce a solution with a sufficiently high objective value, then the instance size must be bounded by an relatively simple function of the parameters. Therefore it is possible that under these conditions the local search component of the memetic algorithm approximates an exhaustive search, or at least has a greater effectiveness. A definite link of this kind would be an interesting development for both parameterized complexity and memetic algorithmics, above and beyond this application.

Materials and Methods

The complexity analysis of the AH-Cut problem employ standard complexity theory techniques [11,12].

The NCI60 cancer dataset was drawn from the NCI60 anti-cancer drug screening program data [20] and the gene expression data for the cell lines was given by Ross *et al.* [15].

The tissue dataset is drawn from Su *et al.* [17].

The synthetic dataset was generated using the methods of Laan and Pollard [18].

For the comparisons we use Gene Cluster's implementation of *k*-Means [4] (http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm#ctv), Dhillon *et al.*'s Graclus software [9] (http://userweb.cs.utexas.edu/users/dml/Software/graclus.html) and Shi and Malik's implementation of Normalized Cut [1].

All experiments were performed on a Dell laptop with a 1.67GHz processor and 2GB of RAM with the software written in Java.

A memetic algorithm for AH-Cut

We have implemented AH-Cut via a memetic algorithm [21,22]. Memetic algorithms provide a population-based approach for heuristic search in optimization problems. Broadly speaking they combine local search heuristics with crossover operators used in genetic algorithms [23–25]. The essence of our algorithm is similar to the work of Merz and Freisleben [26] for Graph Bi-partitioning. Differences arise from the fact that we need to remove the constraint of equal partitioning of the graph. The method consists of three main procedures: a greedy algorithm for initialization of a set of solutions for AH-Cut (detailed in the parameterized algorithm); a differential greedy crossover for evolution of the population; and a variable neighborhood local search, influenced by Festa *et al.* [27], to improve the newly generated solutions.

We use a ternary tree for population similar to Buriol *et al.* [28] and keep two solutions at each node of this tree. One solution is the best obtained so far at the node, called *pocket solution* and the other one is the *current* solution. Essentially, if we generate a current solution by recombination or local search which is better than the pocket solution, we swap the current solution with the pocket solution. Furthermore, each parent node of the tree must have better pocket solution than its children's pocket solutions. Similar tree structures were previously employed successfully for various combinatorially hard problems [28–30].

Differential Greedy Crossover. We allow a crossover of a parent's pocket solution with a child's current solution to ensure the diversity in the population. All vertices that are contained in the same set for both the parents, are included in the same set in the offspring. Then both sets are filled according to a greedy recombination method similar to the greedy algorithm used for the parameterized algorithm. Suppose, the parent solutions P and Q have the partitions S_P, S'_P and S_Q,S'_Q respectively (after interchanging the sets suitably). Then the starting set S (resp. S') for the offspring is given by the intersection $S_P \cap S_Q$ (resp. $S'_P \cap S'_Q$), with the remainder of the partition calculated greedily.

Local Search. We employ a *variable-neighborhood search (VNS)*, first proposed by Hansen and Mladenovic [31] for a local search in the neighborhood of the new offspring. Contrary to other local search methods, VNS allows enlargement of the neighborhood structure. A k-th order neighbor of a paritition giving a solution S for AH-Cut is obtained by swapping the partite set of k vertices. In VNS, first local search is done starting from each neighbor S' of the current solution S. If a solution S'' is found which is better than S, then the search moves to the neighborhood of S''. Otherwise, the order k of the neighborhood is increased by one, until some stop criterion holds. We use maximum value of $\frac{1}{7}\cdot|V(G)|$ for k.

Diversity. Whenever the population stagnates (fails to improve the objective value), we keep the best solution and re-initialize the rest of solutions (using the greedy algorithm with a randomised starting vertex pair) in the set and run the above process again for certain number of generations (say, 30).

To get the optimal solution for very small sized problems (graphs containing less than 25 vertices), we used backtracking. Notice that even though backtracking gives us an optimal solution, a greedy or memetic algorithm may not. By applying this method (backtracking, memetic or greedy algorithm depending on the number of vertices) recursively, we have at each step a graph as input, and the two subgraphs induced by each of the sets of the vertex partition as output; stopping when we arrive to a graph with just one vertex, we generate a hierarchical clustering in a top-down fashion.

Author Contributions

Conceived and designed the experiments: RR, P. Mahata, LM, P. Moscato. Performed the experiments: RR, P. Mahata, LM. Analyzed the data: P. Mahata, P. Moscato. Contributed reagents/materials/analysis tools: RR, P. Mahata, LM. Wrote the paper: P. Mahata, LM.

References

1. Shi J, Malik J (2000) Normalized cuts and image segmentation. IEEE Transactions of Pattern Analysis and Machine Intelligence 22: 888–905.
2. Wertheimer M (1938) Laws of organization in perceptual forms (partial translation). A Sourcebook of Gestalt Psychology. pp 71–88.
3. Papadimitriou CH, Yannakakis M (1991) Optimization, approximation, and complexity classes. Journal of Computer and System Sciences 43: 425–440.
4. de Hoon MJ, Imoto S, Nolan J, Miyano S (2004) Open source clustering software. Bioinformatics 20: 1453–1454.
5. Wu Z, Leahy R (1993) An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. IEEE Transactions of Pattern Analysis and Machine Intelligence 15: 1101–1113.
6. Wang S, Siskind JF (2003) Image segmentation with ratio cut. IEEE Transactions of Pattern Analysis and Machine Intelligence 25: 675–690.
7. Wei YC, Cheng CK (1991) Ratio cut partitioning for hierarchical designs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 10: 911–921.
8. Yu SX, Shi J (2003) Multiclass spectral clustering. In: International Conference on Computer Vision.
9. Dhillon IS, Guan Y, Kulis B (2007) Weighted graph cuts without eigenvectors: a multilevel approach. IEEE Transactions of Pattern Analysis and Machine Intelligence 29: 1944–1957.
10. Mahata P, Costa W, Cotta C, Moscato P (2006) Hierarchical clustering, languages and cancer. In: EvoWorkshops 2006. pp 67–78.
11. Ausiello G, Crescenzi P, Gambosi G, Kann V, Spaccamela AM, et al. (1999) Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties. New York: Springer-Verlag.
12. Flum J, Grohe M (2006) Parameterized Complexity Theory. New York: Springer.
13. Karp RM (1972) Reducibility among combinatorial problems. In: Proceedings of a symposium on the Complexity of Computer Computations. New York: IBM T. W. Watson Research Center. pp 85–104.
14. Mahajan M, Raman V (1997) Parameterizing above guaranteed values: Maxsat and maxcut. Electronic Colloquium on Computational Complexity (ECCC) 4.
15. Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, et al. (2000) Systematic variation in gene expression patterns in human cancer cell lines. Nature Genetics 24: 227–235.
16. Yap YL, Zhang XW, Danchin A (2003) Relationship of SARS-CoV to other pathogenic RNA viruses explored by tetranucleotide usage profiling. BMC Bioinformatics 4: 43.
17. Su AI, Cooke MP, ching KA, Hakak Y, Walker JR, et al. (2002) Large scale analysis of the human and mouse transcriptomes. PNAS 99: 4465–4470.
18. Laan MJ, Pollard KS (2003) A new algorithm for hybrid clustering of gene expression data with visualization and bootstrap. Journal of Statistical Planning and Inference 117: 275–303.
19. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, et al. (1999) Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286: 531–537.
20. NCI/NIH (2010) Developmental Theraputics Program. Available: http://dtp.nci.nih.gov/.
21. Moscato P (1989) On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical Report 826, Caltech Concurrent Computation Program.
22. Moscato P, Cotta C (2003) A gentle introduction to memetic algorithms. In: Glover F, Kochenberger G, eds. Handbook of Metaheuristics. Boston MA: Kluwer Academic Publishers. pp 105–144.
23. Whitley D (1994) A genetic algorithm tutorial. Statistics and Computing 4: 65–85.
24. Moscato P, Cotta C, Mendes A (2004) Memetic algorithms. In: Onwubolu G, Babu B, eds. New Optimization Techniques in Engineering. Berlin: Springer-Verlag. pp 53–85.
25. Moscato P, Cotta C (2006) Memetic algorithms. In: González T, ed. Handbook of Approximation Algorithms and Metaheuristics, Taylor & Francis.
26. Merz P, Freisleben B (2000) Fitness landscapes, memetic algorithms, and greedy operators for graph bipartitioning. Evolutionary Computation 8: 61–91.
27. Festa P, Pardalos P, Resende MGC, Ribeiro CC (2002) Randomized heuristics for the MAX-CUT problem. Optimization Methods and Software 7: 1033–1058.
28. Buriol L, Franca PM, Moscato P (2004) A new memetic algorithm for the asymmetric traveling salesman problem. Journal of Heuristics 10: 483–506.
29. Berretta R, Cotta C, Moscato P (2004) Enhancing the performance of memetic algorithms by using a matching-based recombination algorithm. In: Metaheuristics: computer decision-making. Norwell, MA, USA: Kluwer Academic Publishers. pp 65–90.
30. Mendes A, Cotta C, Garcia V, Franca P, Moscato P (2005) Gene ordering in microarray data using parallel memetic algorithms. In: ICPP Workshops. pp 604–611.
31. Hansen P, Mladenović N (2001) Developments of variable neighborhood search. Essays and Surveys in Metaheuristics. pp 415–439.