cuACS Project

Revised Algorithm Design Document

Team 070 – Segmentation Fault

Weihang Chen

Liyongshi Chen

Richard Xia

Xiran Zhou

Submitted to:

Dr. Christine Laurendeau

COMP3004 Object-Oriented Software Engineering

School of Computer Science

Carleton University

Table of Content:

1. Introduction

1.1. Overview of the Algorithm

The design goal of the animal-client matching (ACM) algorithm is to generate an optimal set of animal-client matches, for which clients' requirements and preferences should be satisfied as much as possible with more sheltered animals being adopted. To achieve this goal, we constructed our ACM algorithm into two sub algorithms:

1. Profile matching (begin algorithm), where points are calculated using a set of rules to determine the compatibility of each animal-client pair.

2. Adoption maximization (end algorithm), where a sorted list is established according to the number of possible matches for each client, as well as for each animal available for adoption.

The ACM algorithm should output a set of animal-client pairs, such that the best interests of all sheltered animals are considered in addition to clients' desires.

1.2. Overview of Document

The purpose of this document is to describe our ACM algorithm design in such a way that it is understood by all interested parties. This document is divided into multiple sections including an introduction, the animal physical attributes used by the algorithm and their valid range, the rules for matching, the way to compute the optimal set of matches, and a summary of the document, respectively.

1.3. Revision

There are several major changes from the previous document in terms of the algorithm design:
Profile Matching Algorithm:

1. We added rules for animal physical attribute so that the physical attributes are considered in the matching.
2. We added 3 sub matching stages and they are executed in sequence:
Prime Matching -> Client Preference Matching -> Inferred Attribute Matching;
3. We amended our general formula;
4. We also amended our detailed formula to some matching rules as well as their points;

There are also some minor changes in the document: we added more verifications of the attributes that are referenced in our rules, so that each rule is justified more thoroughly; we also added more explanations and descriptions to our algorithm design, in order to make the document more readable.

2. Animal Attributes

2.1. Non-physical Attributes

Non-physical attributes of an animal, i.e. temperament and habits, are traits that cannot be measured directly (Unlike the weight of an animal). The non-physical attributes used in the ACM algorithm as well as their valid range of values are listed in Table 1 below.

Table 1 – Non-physical Attributes Used to Compute Animal and Client Matches

Non-physical attributes	Valid Range of Values	Relation to Clients
1. Aggressivity • Testy • Middle • Mild	Based on prime rules.	Inferred from client's age and family member information.
2. Personality • Active • Quiet • Shy • Lazy • Friendly	{1/5, 2/5, 3/5, 4/5, 1}, with 1/5 being the least desirable match and 1 being the most desirable match. Each value of this attribute worth 1/5 point.	Implied by client's choice. Clients can choose as many values as they want for this attribute.
3. Sociality • Gregarious • Solitary • Gregarious/solitary	{1, 2, 3}, with 1 being the least desirable match and 3 being the most desirable match.	Inferred from client's daily schedule, i.e., how many hours the client can spend on taking care of pets.
4. Food Preferences • Low sugar • Low salt • Uncooked • Low fat • Only Liquid	$\{x \in R \mid 0 \leq x \leq 1\}$, with 0 being the least desirable match and 1 being the most desirable match.	Inferred from client's income. More values chosen in this field meaning the more cost on animal food is going to be.
5. Service Ability • Guide dog • Animal therapy • Children care • None	{0, 1}, with 0 being the least desirable match and 1 being the most desirable match. This attribute is chosen by the client.	Implied by client's choice.

6. Interaction ● Simple gesture commands ● Simple verbal commands ● Complex verbal commands ● None	{0, 1/4, 2/4, 3/4, 1}, with 0 being the least desirable match and 1 being the most desirable match.	Inferred from client's age and family member information.
7. Psychological Disorder ● Accident ● Abandoned ● Maltreated ● None	{0,1}based on the prime rule	Inferred from client's years of animal care experience.
8. Activity Habit ● Active at daytime ● Active at night ● Never active ● Always active	{0, 2/5, 3/5, 1}, with 0 being the least desirable match and 1 being the most desirable match.	Inferred from client's daily schedule.
9. Days Since Sheltered ● Integer of days	$\{x \in R \mid 0 \leq x \leq 1\}$, with 0 being the least desirable match and 1 being the most desirable match.	Calculated by formula.
10. Adaptability ● Good ● Need some time ● Need a long time	{1, 2, 3}, with 1 being the least desirable match and 3 being the most desirable match.	Inferred from client's daily schedule, i.e., how many hours the client can spend on pets.
11. Resource ● Previous owners ● Homeless ● Laboratory ● Rescued	Based on prime rule	Inferred from client's animal care experience.
12. Necessities Needs ● 0-200 ● 200-400 ● 400-600 ● More that 600	$\{x \in R \mid 0 \leq x \leq 1\}$, with 0 being the least desirable match and 1 being the most desirable match.	Inferred from client's income.

2.2. Physical Attribute

Physical attributes of an animal, i.e. species, ages, and weights, are characteristics that can be measured or visualized directly. They are mostly determined by genetic traits by the parent animal. The physical attributes used in the ACM algorithm as well as their valid range of values are listed in Table 2 below.

Table 2 – Physical Attributes Used to Compute Animal and Client Matches

Physical attributes	Valid Range of Values	Relation to Clients
1. Age	{0,0.5,1}	Implied by client's choice.
2. Species	{0,1}	Implied by client's choice.
3. Disability	{0,1}based on the prime rule	Implied by client's choice.

3. ACM Matching Rules

3.1. Overview of the ACM Matching Rules

This section will focus on the begin algorithm of the ACM – Profile Matching sub algorithm. It implements a unique set of rules to compute animal-client matches. The rules have been defined as four unique types, which are prime rules, preference rules, equations and the distribute rules.

Assume the total value of prime rules as P,

the total value of preference rules as R,

the total value of equations as E

and the total value of distribute rules as D.

The final equation would be

S = final matching grade of all subsystems

S = P * (R+E+D)

3.2. Profile Matching Sub Algorithm

Table 3 – Animal Attributes and Associated Client Information Dependencies

Non-physical Attribute	**Associated Client Information**
1. Aggressivity	Has children; Age
2. Personality	Preferences
3. Sociality	Hours willing to spend on pets per day
4. Food Preferences	Monthly income
5. Service Ability	Special requirements
6. Interaction	Has children; Age
7. Psychological Disorder	Animal care experience
8. Activity Habit	Daily routine
9. Days Since Sheltered	N/A
10. Adaptability	Hours willing to spend on pets per day
11. Resource	Animal care experience
12. Necessities Needs	Monthly income

Rules for subsystem and their detailed description

Based on the description of the matching subsystem, we have 4 types of rules to follow.

1. Prime

the prime rules consider the specific groups which have a special need which will have a huge impact on the success of the match.It is helpful to avoid generating excessive and unreasonable matches.

.

The prime has two parts which are

1) family

2) animal special care

Use the formula to exclude animals that do not meet the conditions. If any condition doesn't match, the value will be assigned as 0 directly

P = $\prod_n f(n)$

1) Family

The family contains children or elder people. The reason why we focus on the group is children and elder people need to more looking after. We thought the animal they adopt should more kind and friendly. To approach it, we consider the aggressively as animal's attribute considered.

If the client has children or elder people as the family member

The f(1) will be marked as 0 if the animal is testy (to avoid matching the testy animal to those clients)

The f(1) will be marked as 0.5 if the animal has mid aggressivity when either elder or kid in their family

Otherwise,

if the client doesn't have family member who is kid or elder people,

all of the value is marked as 1 (the aggressively will not be a conclusive factor)

2) Animal special care

The shelter has a high possibility to receive animals which have disability or trauma(both side of physical and mental). Those group of animals needs more attention from their owners. The owner has to spend a lot of effort into animals. We prefer to let the clients make the decision by themselves whether be willing to take care of those special group.

If the client is unwilling to offer special care

All the value has been marked as 0 if the animal has disability, psychology disorder or surviving from lab

Otherwise

The client can choose which type of special animal he/she wants to adopt

Those group of animal matched pervious requirements(such as disability) will be marked as 1

2. Preference
According to the client personal preference, the client can choose some animal's attributes what is in line with their wishes. This section reflects the explicit requirements of the user.

The formula for total number:
$R = \sum_{n} r(n)$
Eligible attributes are:
1) personality
2) animal age
3) classification
4) special service

1) Personality (multiple selections)
The client can choose the animal's personality; each pair of matching will add points as

Personality: r(1) = (1/5)*A.persPoints		
Attr. value	*Rule*	*Reason for the rule*
Active	Each value worth 1 point; Client can choose more than one value; A.persPoints = the total number of choices that a client has selected. Note: r(1)=(1/5)* A.persPoints.	This rule is to consider the preferences of a client and try to give a match that is relatively favorable to the client.
Quiet		
Shy		
Lazy		
Friendly		

2) Animal age (single selection)
The client's animal age preference, such as newborn or mature. The perfect matching values 1 point and ever gap will lead 0.5 points reduction
f(2)= {x| x∈{0,0.5,1}}

3) Classification
Client can choose between different species of animals, animals with the demand classification will get extra 1 point.
f(3) = {x|x∈{0,1}}

4) Special Service

Client can choose special service if they need it, such as mental therapy, service dog
The matching animals will get a large number of points which will beyond the normal boundary.
On the other hand, other animal which is mismatching will be assigned as 0

Considering the service animal is minority, we want them as useful as possible,
if the client doesn't have the need,
the animal which offer special service will marked as -1
Note:
If more than one animal provides services, we also want to find the most suitable object for the client. That is why we do not set it as a prime attribute

3. Equation

Based on the information the client provided and the attribute the animal had, the equation rules use mathematical formulas to quantify data . By analyzing the implicit associations of information, the rules can help the matching result to be more logical and rational.

$$E = \sum_{n} e(n)$$

1) Diet

The diet condition will associate with the client's economy ability. High quality food always has expensive price.

Attr. value	*Rule*	*Reason for the rule*
Low Sugar	s =client's salary; n = number of selected choices. e1 = min {1, ((s – 2000)/500) / n}	The more selections a client makes in Food Preferences, the more cost it is to feed this animal. This rule is to make sure that a client is capable of feeding the animal.
Low Salt		
Uncooked		
Low fat		
Only Liquid		

2) Day Since Sheltered

Attr. value	*Rule*	*Reason for the rule*
Integer of days	If A.day_since_sheltered<15 Then e2= 0; Else e2 = min{1, ln((day-15) /5+1) }	First in first out. When an animal spends more than 15 days in the shelter, it should be considered first for adoption.

3) Necessary Investment

Income_index: section of investment ability;

N =min{3, floor((salary/2000)}

e3 = 1 – abs(income_index - N)*(1/3)

Attr. value	*Income_index*	*Rule*	*Reason for the rule*
0-200	1	Income_index: investment ability; N =min{3, floor((salary/2000)} e3 = 1 – abs(income_index - N)*(1/3)	Some animals may require much more money on necessities than others. Clients who are able to afford the expense are matched with high-cost animals.
200 - 400	2		
400 -600	3		
> 600	4		

4.Distribution

Parts that are implicitly related but cannot be formulated will be assigned values. The distribution rules give different distributions to attributes based on the situations.

$$D = \sum_{n} d(n)$$

1) Sociality

t= number of hours available for pets (hours/day); Possible values for t: 2, 4, 6 (hours/day).		
Attr. value	*Rule*	*Reason for the rule*
Gregarious	If t==2 then d(1) =1/3; If t==4 then d(1)=2/3; If t==6 then d(1)=3/3;	This rule tries to match solitary animals to clients that have busy schedules and gregarious animals to less busy clients.
Solitary	If t==2 then d(1)=3/3; If t==4 then d(1)=2/3; If t==6 then d(1)=1/3;	
Gregarious or Solitary	If t==2 then d(1)=2/3; If t==4 then d(1)=3/3; If t==6 then d(1)=2/3	

2) Adaptability

t= number of hours available for pets (hours/day); Possible values for t: 2, 4, 6 (hours/day).		
Attr. value	*Rule*	*Reason for the rule*
Good	If t==2 then d(2)=3/3; If t==4 then d(2)=1/3; If t==6 then d(2)=1/3;	For clients who don't have much free time to take care of their pets, animals with good adaptabilities are likely to be a good match. For clients who are able to spend more time on their pets, less
Need some time	If t==2 then d(2)=1/3; If t==4 then d(2)=3/3; If t==6 then d(2)=2/3;	

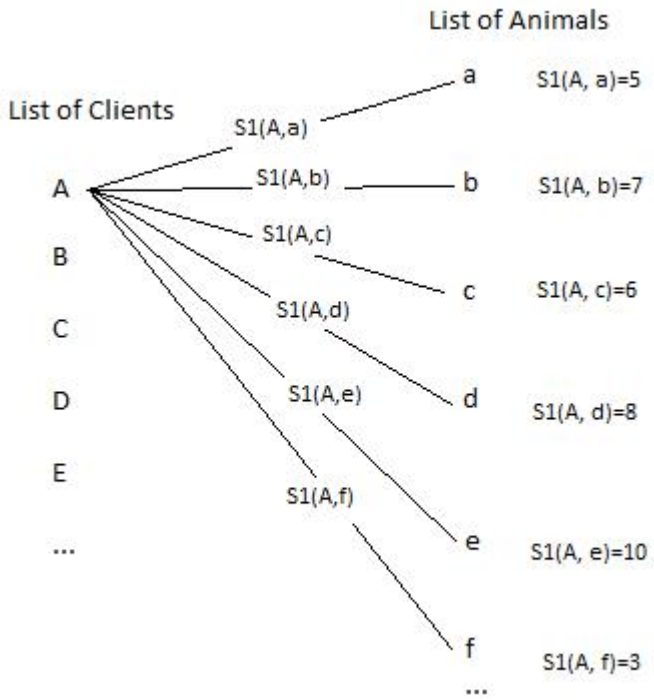
Need a long time	If t==2 then d(2)=1/3; If t==4 then d(2)=2/3; If t==6 then d(2) =3/3;	adaptive animals are more likely to be assigned.

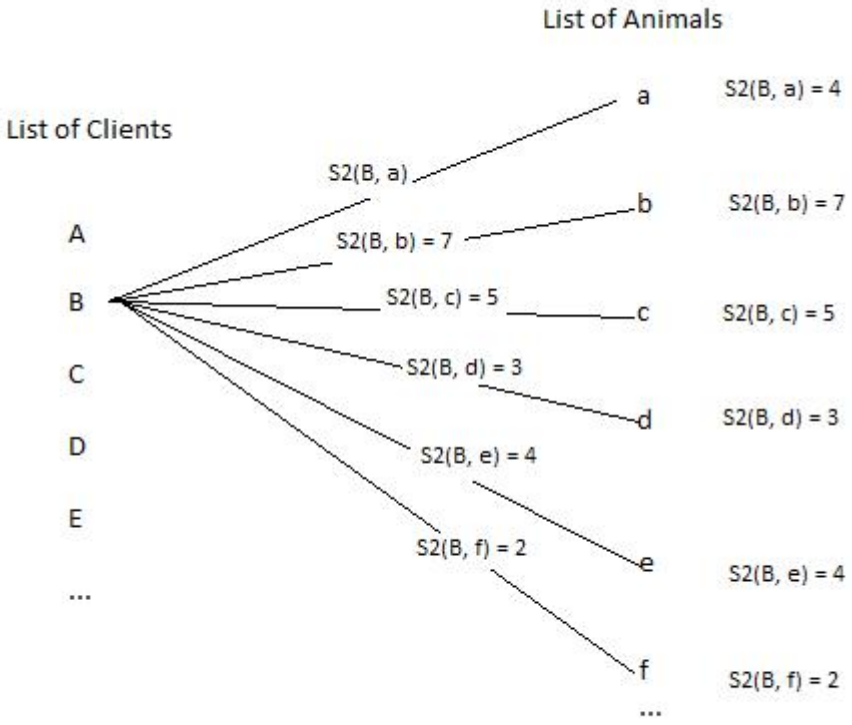
Graph 1 – Flow of the Profile Matching Sub Algorithm

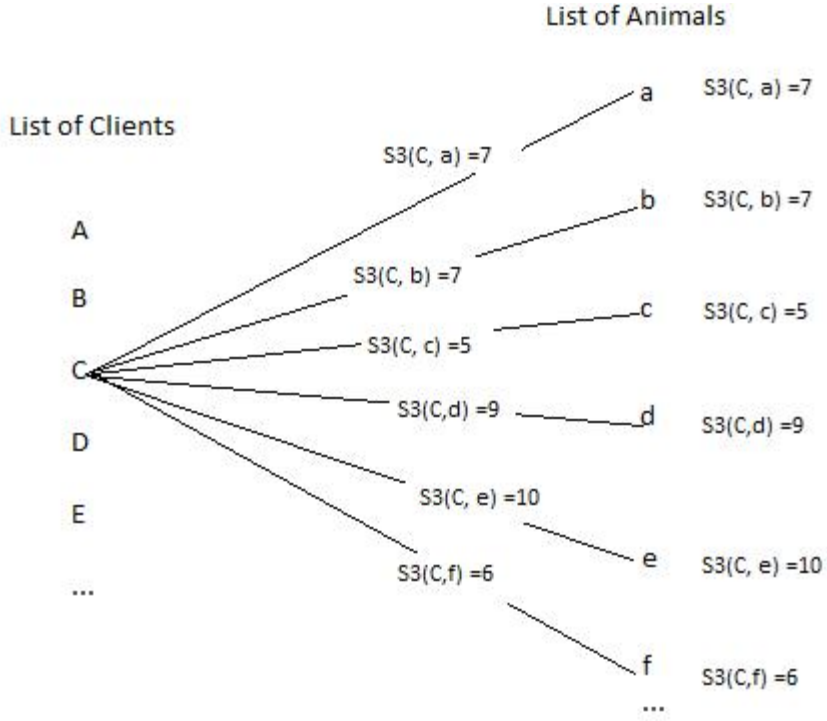
A sequence of graphs is constructed in this subsection to show how the Profile Matching algorithm is executed by giving a simple example. The execution steps are as follows: first, a list of Clients l_client = {A, B, C, D, E, ...} is extracted from the Human Information Database System and a list of Animals l_animal = {a, b, c, d, e, ...} is extracted from the Animal Information Database System. Second, each Client element l_client[i] from the Client List traverses the Animal List and compute S_i. Once all clients traversed the Animal List, select client-animal pairs (l_client[i], l_animal[i]) where S_i(l_client[i], l_animal[i])>6. This set of pairs is defined as the output of the Profile Matching algorithm.

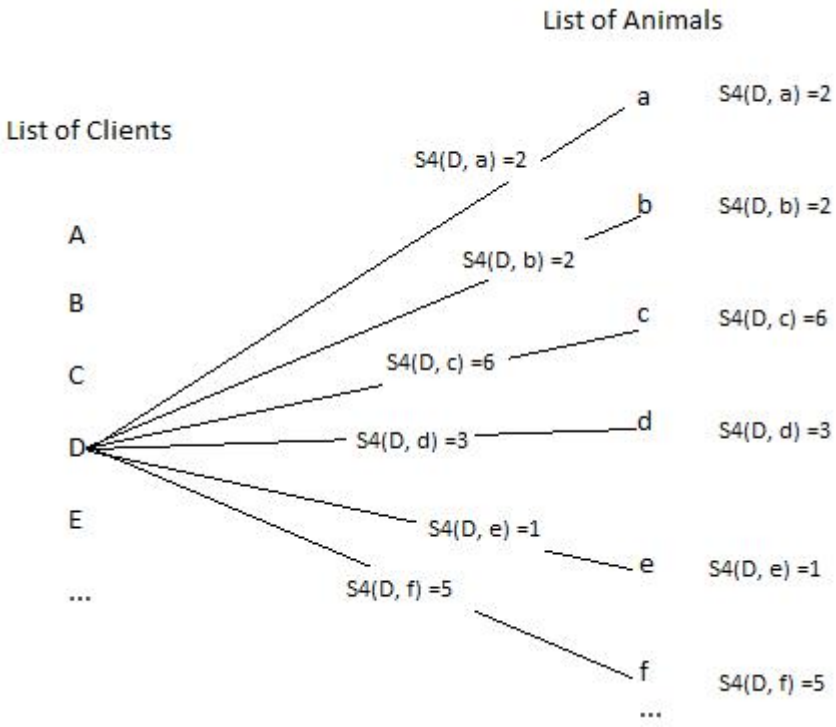
The sequence of graphs below illustrates the flow of Profile Matching algorithm execution.

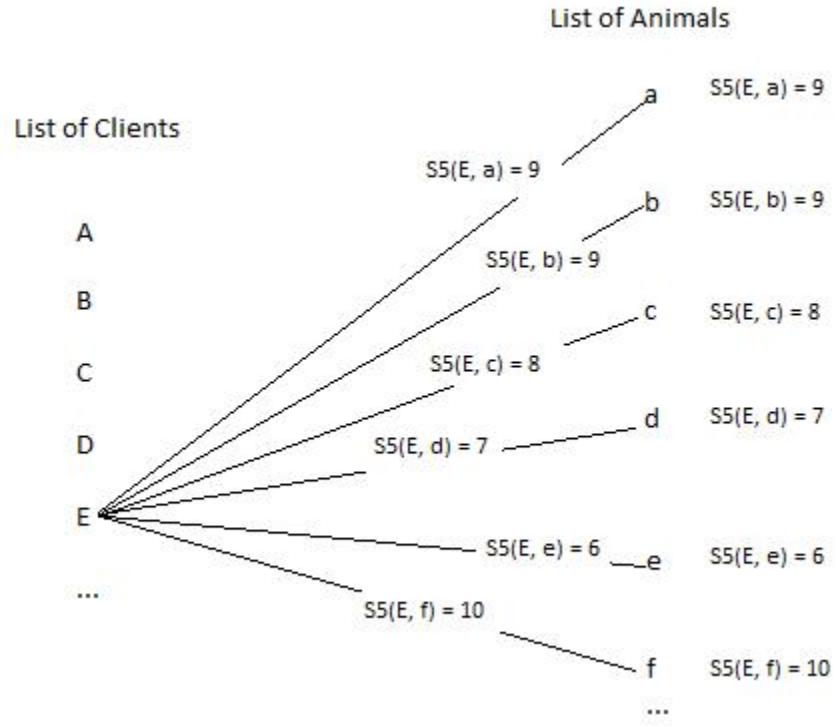
Stage 1: Each client *i* traverses the list of animals and compute point subtotal S_i.




List of Animals
List of Clients
A
B
C
D
E
...
S2(B, a)
S2(B, b) = 7
S2(B, c) = 5
S2(B, d) = 3
S2(B, e) = 4
S2(B, f) = 2
a
b
c
d
e
f
...
S2(B, a) = 4
S2(B, b) = 7
S2(B, c) = 5
S2(B, d) = 3
S2(B, e) = 4
S2(B, f) = 2


List of Animals
List of Clients
A
B
C
D
E
...
S3(C, a) =7
S3(C, b) =7
S3(C, c) =5
S3(C,d) =9
S3(C, e) =10
S3(C,f) =6
a
b
c
d
e
f
...
S3(C, a) =7
S3(C, b) =7
S3(C, c) =5
S3(C,d) =9
S3(C, e) =10
S3(C,f) =6


List of Animals
List of Clients
A
B
C
D
E
...
a
b
c
d
e
f
...
S4(D, a) =2
S4(D, b) =2
S4(D, c) =6
S4(D, d) =3
S4(D, e) =1
S4(D, f) =5
S4(D, a) =2
S4(D, b) =2
S4(D, c) =6
S4(D, d) =3
S4(D, e) =1
S4(D, f) =5


List of Animals
List of Clients
A
B
C
D
E
...
a
b
c
d
e
f
...
S5(E, a) = 9
S5(E, b) = 9
S5(E, c) = 8
S5(E, d) = 7
S5(E, e) = 6
S5(E, f) = 10
S5(E, a) = 9
S5(E, b) = 9
S5(E, c) = 8
S5(E, d) = 7
S5(E, e) = 6
S5(E, f) = 10

Stage 2: Select pairs which satisfy S > 6.

Client	Matched Set of Animals	Number of Matches
A	{b, d, e}	|A| = 3
B	{b}	|B| = 1
C	{a, b, c, e}	|C| = 4
D	{ }	|D| = 0
E	{a, b, c, d, f}	|E| = 5

Animal	Matched Set of Clients	Number of Matches
a	{C, E}	|a| = 2
b	{A, B, C, E}	|b| = 4
c	{E}	|c| = 3
d	{A, C, E}	|d| = 3
e	{A, C}	|e| = 2
f	{E}	|f| = 1

Stage 3: Generate a set of client-animal pairs as the output of this sub algorithm.

The matrix below shows the resulting client-animal pairs, where 1 stands for a valid pair (i.e., S>6) and 0 stands for an invalid pair (i.e., S<=6). Cells that have a value of 1 will be extracted in pairs in the form of (client_i, animal_i) and put into a set R as the output. The set R is then used as input for the next algorithm – Adoption Maximization.

Result:

Client\Animal	a	b	c	d	e	f
A	0	1	0	1	1	0
B	0	1	0	0	0	0
C	1	1	0	1	1	0
D	0	0	0	0	0	0
E	1	1	1	1	0	1

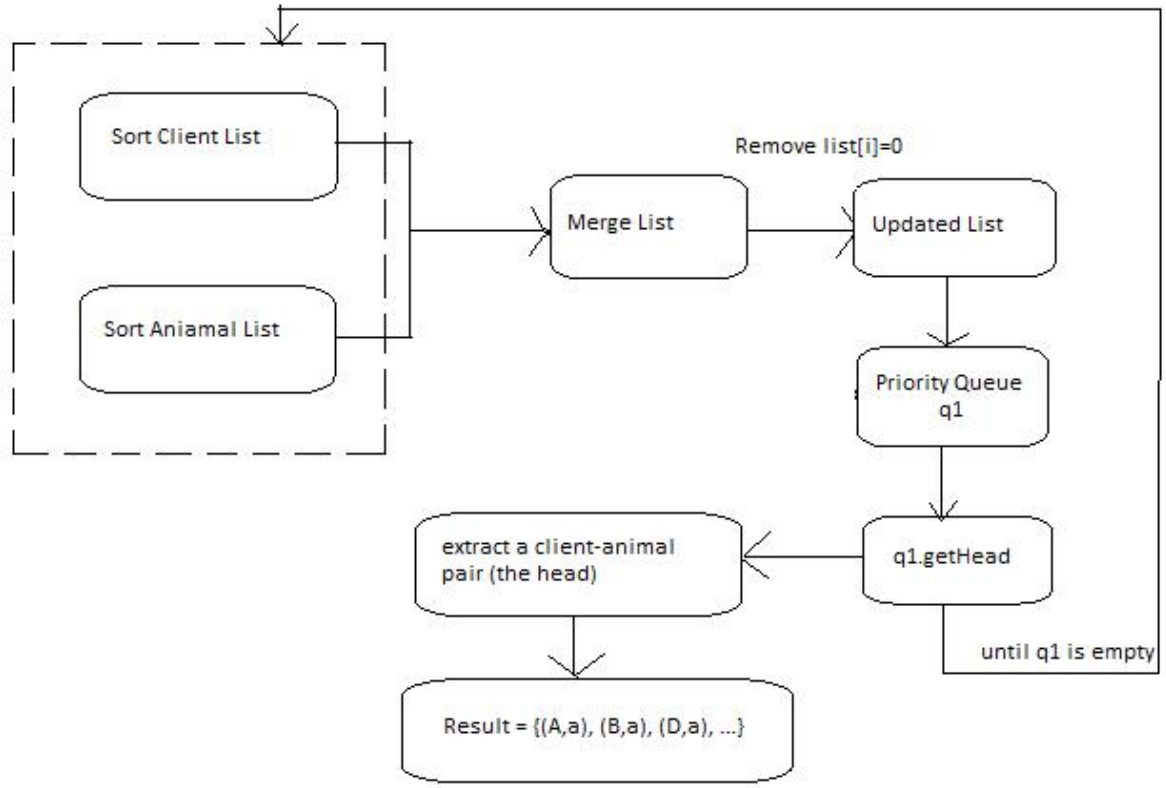
4. Computing Optimal Set of Matches

This section explains how the optimal set of matches are computed using the combination of Profile Matching sub algorithm and Adoption Maximization sub algorithm. First, the Adoption Maximization sub algorithm will be introduced and explained, then a detailed description will be provided stating how the ACM matching rules are used together with the Adoption Maximization sub algorithm to generate the final output of ACM algorithm.

4.1. Adoption Maximization Sub Algorithm

Adoption Maximization sub algorithm aims to take the best interests of all sheltered animals while producing relatively favourable matches to the clients. This sub algorithm can be referred to as the end algorithm. It takes the result from the previous algorithm (begin algorithm) and outputs a set of optimized client-animal matches in the end. The graph below illustrates the execution sequence of the Adoption Maximization sub algorithm.

Graph 2 – Flow of the Adoption Maximization Sub Algorithm



Stage 1: In the previous section (section 3-3.2), the cardinality of each list element in the result – number of matches, is computed during *stage 2*. Sort their cardinality from smallest to largest and put them into list_1.

For Clients				
|D|	|B|	|A|	|C|	|E|
0	1	3	4	5

Therefore, list_1 = [“D”, “B”, “A”, “C”, “E”].

Similarly, sort the animal list using the same principle, and put them into list_2.

For Animals					
|c|	|f|	|a|	|e|	|d|	|b|
1	1	2	2	3	4

Therefore, list_2 = [“c”, “f”, “a”, “e”, “d”, “b”].

Then, merge the two sorted lists into a single list list_3.

|D|	|c|	|f|	|B|	|a|	|e|	|d|	|A|	|C|	|b|	|E|
0	1	1	1	2	2	3	3	4	4	5

Therefore, list_3 = [“D”, “c”, “f”, “B”, “a”, “e”, “d”, “A”, “C”, “b”, “E”].

Stage 2: From list_3, delete list elements that hold a value of 0. This action is to remove clients/animals that have no valid matches from the list.

|c|	|f|	|B|	|a|	|e|	|d|	|A|	|C|	|b|	|E|
1	1	1	2	2	3	3	4	4	5

Therefore, updated list_3 = [“c”, “f”, “B”, “a”, “e”, “d”, “A”, “C”, “b”, “E”].

Stage 3: Build a priority queue based on list_3, where list elements that hold the smallest value have top priority. As a result, Animal c has the top priority, and then Animal f has the second priority, and Client B has the third priority.

Clients or Animals that have the top priorities are considered first for matching (FIFO). Therefore, the head of the priority queue is extracted as a pair of matches. After the head is popped from the priority queue, update all lists and queues that were previously stated.

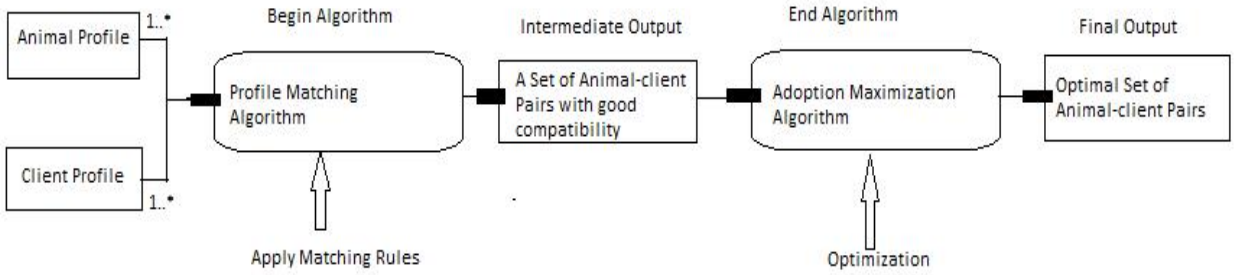
Stage 4: Keep repeating the operations of *Stage 2* and *Stage 3* until the priority queue is empty. Put all valid client-animal pairs generated by the previous matching algorithm into a set R.

Stage 5: Output set R as the final result of the ACM algorithm.

4.2. Combining Matching Rules and Optimization

As previously stated, the two sub algorithms work together to provide the full feature of the ACM algorithm. Profile Matching uses a unique set of rules to ensure the animal-client pairs are compatible and gives the basic matching result. Adoption Maximization optimizes the matching result from the previous algorithm and gives a set of recommended output matches to the users. The latter algorithm aims to benefit as many cuACS participants as possible, including human clients and sheltered animals.

The high-level block diagram below shows the flow of execution of the ACM algorithm. Combining matching rule together with optimization method enables the system to perform at full-fledged functionality:



5. Summary

The ACM algorithm is the core of the cuACS software system. Two sub algorithms – Profile Matching and Adoption Optimization, are established and utilized to produce the full functionality of ACM algorithm. With a well-designed Profile Matching algorithm, the cuACS will find a nice home for sheltered animals intelligently and fulfill the needs of the clients effectively. Once the Adoption maximization algorithm is added, the cuACS will be optimized even more such that the largest possible group of clients as well as the largest possible group of animals would benefit from the matching result of the cuACS software system.