Heuristic Design of Property Maps

Ravi Darira Karen C. Davis Jennifer Litton
Electrical & Computer Engineering Dept.
University of Cincinnati
Cincinnati, OH USA

UNIVERSITY OF Cincinnati

Overview

- introduce Property Map
- size of solution space
- background results
  - initial design algorithms
  - performance study
- selection of heuristics to implement
- design algorithms
  - implementation, demonstration, evaluation
- next steps

Property Map

- A property map is composed of property strings
- a property string is a bit string that encodes
  - Boolean predicates
  - ranges
  - enumerated values

Boolean property
destination == "New York"

flightno	airfare	destination	P1	P2	P3	Pstring
101	220	New York	00	01	1	00011
257	140	Cincinnati	10	00	0	10000
424	400	Troy	10	11	0	10110

range property
on *flightno*

Enumerating the Possible PMaps

If we have 3 predicates over the same attribute:

boolean over predicate 2 →

B1	R1-3	B1,R2-3
B2	B1,B2	R1-2,B3
B3	B2,B3	R1-3,B2
R1-2	B1,B3	R1-2-3
R2-3	B1,B2,B3	E1-2-3

← range over predicates 1 and 2 and boolean over predicate 3

15 possible combinations

↑ only one enumerated property over all predicates

How Big Is the Solution Space?

- *n* = number of predicates over the same attribute
  - number of Boolean properties for *n*

$$B = 2^n - 1$$

  - number of range properties for *n*

$$R = \sum_{i=2}^{n} \binom{n}{i}$$

  - number of combinations of Boolean, range, and enumerated for predicates for *n*

$$T = B + R + (B*R) + 1$$

- for *k* predicate groups (*k* different attributes)

$$\prod_{i=1}^{k} T_i$$

UNIVERSITY OF Cincinnati

Allowing Multiple Ranges within a Group

Instead of just one range property over a set of predicates for the same attribute, allow different ranges:

R1-2,R3-4,R5-6 … this is not the same as R1-2-3-4-5-6!

Turns out to be a Bell number: the number of distinct partitions of *n* items

$$B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k.$$

substitutes for *R* on previous slide

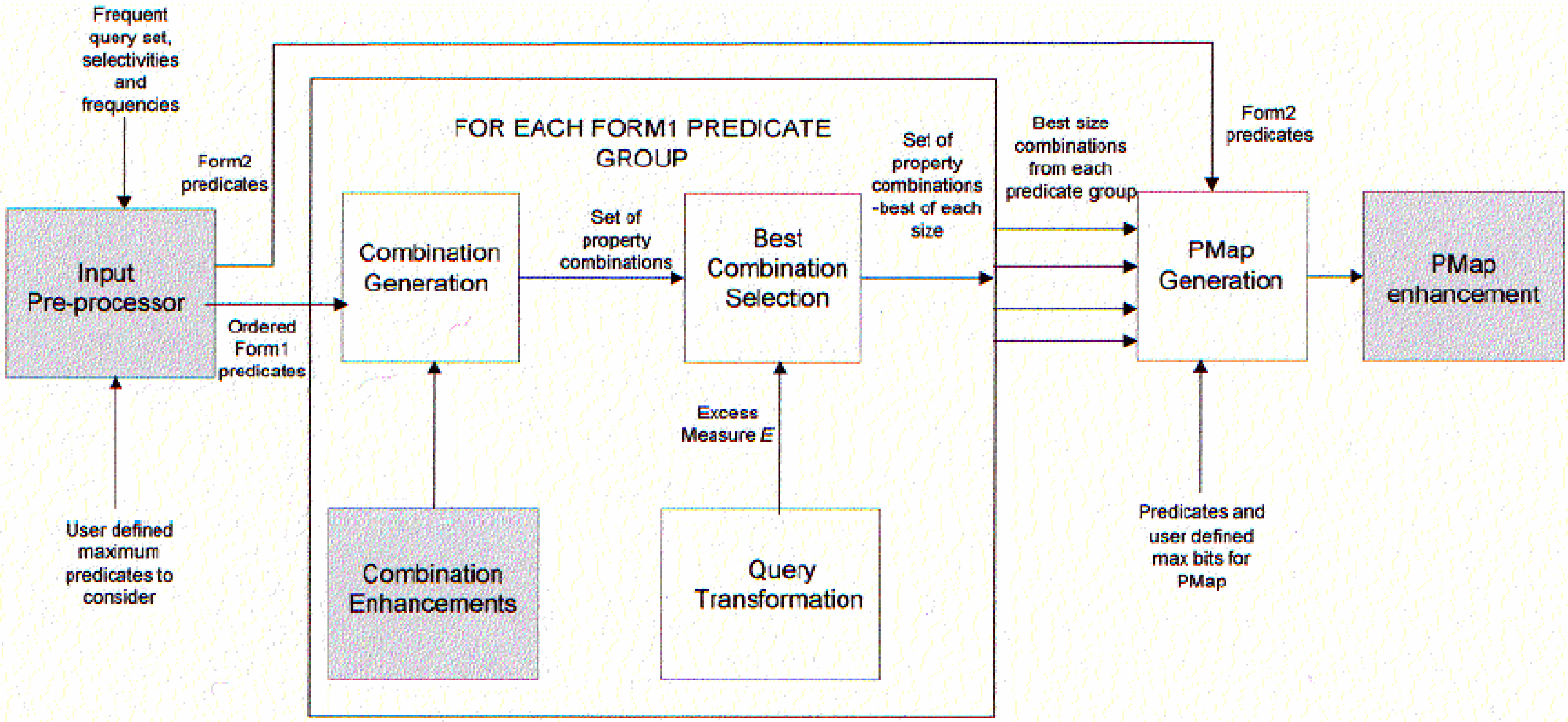
http://wikipedia.org/wiki/Bell_numbers

UNIVERSITY OF Cincinnati

Challenges for a PMap Design Algorithm

- selecting the predicates to cover
- selecting the types of properties for those predicates
- generating property combinations
- selecting the best combinations to form a PMap
- determining a measure for best

Design Algorithms: Implementation



Design Algorithms: Combination Generation

No.	Predicate	Group
2	airfare < 300	airfare
5	airfare < 501	airfare
11	airfare < 600	airfare
7	airfare < 700	airfare
1	alcode < 100	alcode
4	alcode < 150	alcode
6	alcode < 120	alcode
10	alcode < 130	alcode
8	stopovers < 3	stopovers
9	stopovers < 4	stopovers
3	stopovers < 2	stopovers

preprocessing separates predicates, estimates selectivity, transforms to “<”

Generate a combination on each iteration that provides better coverage

- modify range end points
- combinations should not exceed size of enumerated property

airfare only

Combination	Bits
B2	1
R2-5	2
B2,B5	2
R2-5-11	2
B2,R5-11	3
R2-5,B11	3
B2,B5,B11	3

Queries	Selectivity	Frequency
alcode < 100 AND airfare < 300 AND destination == "Cincinnati" AND stopovers < 2	0.0056	1000
alcode < 150 AND airfare > 500 AND destination == "New York"	0.0075	2000
alcode < 120 AND airfare < 700 OR destination == "Boston" AND stopovers < 3	0.0040	2500
alcode == 110 OR airfare == 400 AND destination == "Cincinnati" AND stopovers < 4	0.0030	1500
alcode < 130 OR airfare < 600 AND destination == "Tampa" AND stopovers == 3	0.0025	800

Combination	Bits	E
B2	1	17452
R2-5	2	11107
B2,B5	2	11152
R2-5-11	2	10947
B2,R5-11	3	10992
R2-5,B11	3	17371
B2,B5,B11	3	17371

airfare

Select best coverage at each size

Group	Bits	Combination
airfare	1	B2
	2	R2-5-11
	3	B2,R5-11
alcode	1	B1
	2	R1-6-10
	3	B1,R6-10
stopovers	2	E3-8-9

Design Algorithms:
Property Map Generation

No.	Predicate
2	airfare < 300
5	airfare < 501
11	airfare < 600
7	airfare < 700
1	alcode < 100
6	alcode < 120
14	destination == Boston
3	stopovers < 2
10	alcode < 130
13	destination == New York
4	alcode < 150
16	airfare == 400
15	alcode == 110
12	destination == Cincinnati
8	stopovers < 3
17	destination == Tampa
18	stopovers == 3
9	stopovers < 4

ordered by a function of selectivity and frequency

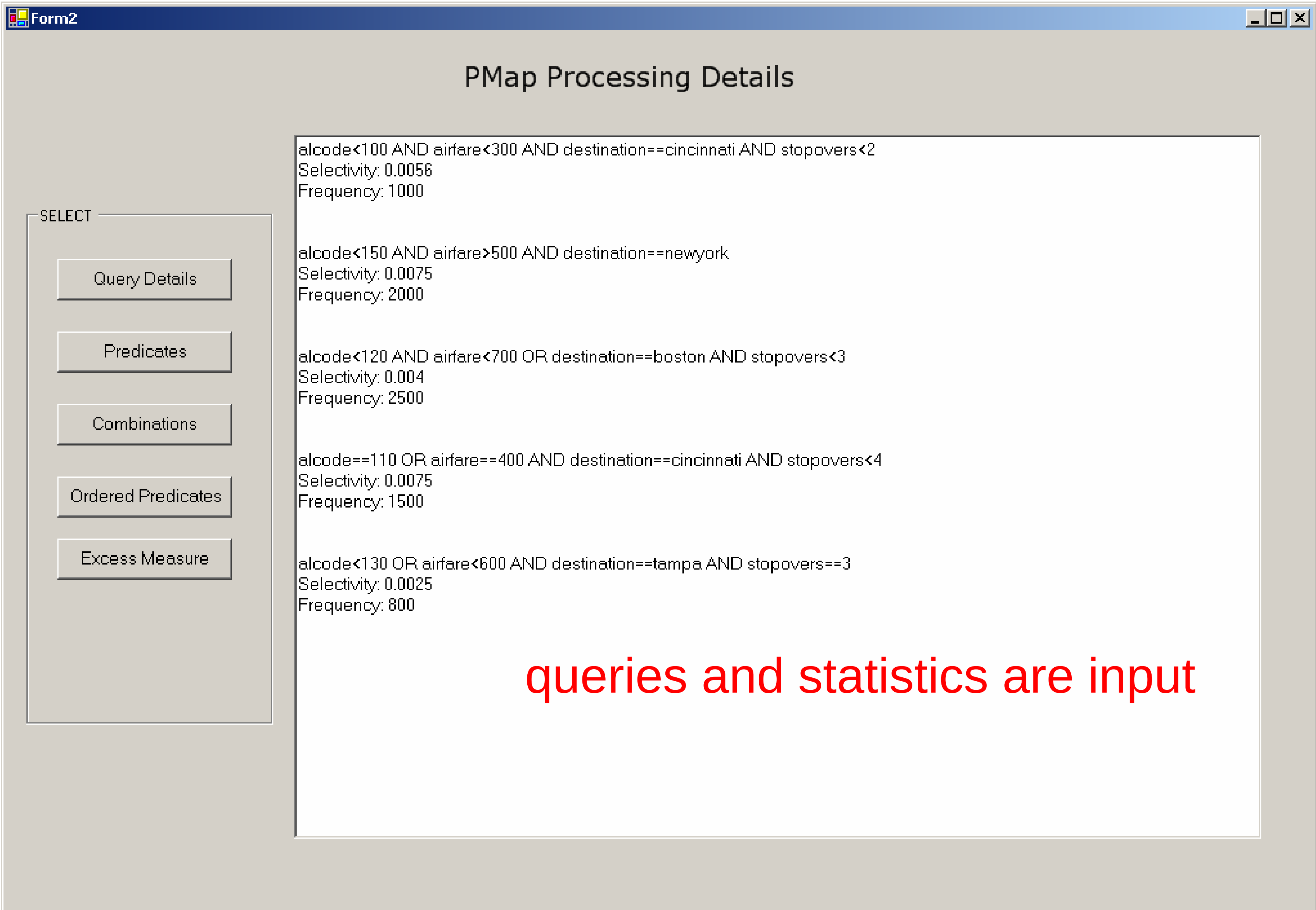
Group	Bits	Combination
airfare	1	B2
	2	R2-5-11
	3	B2,R5-11
alcode	1	B1
	2	R1-6-10
	3	B1,R6-10
stopovers	2	E3-8-9

For each predicate, add a combination that covers it with least number of bits

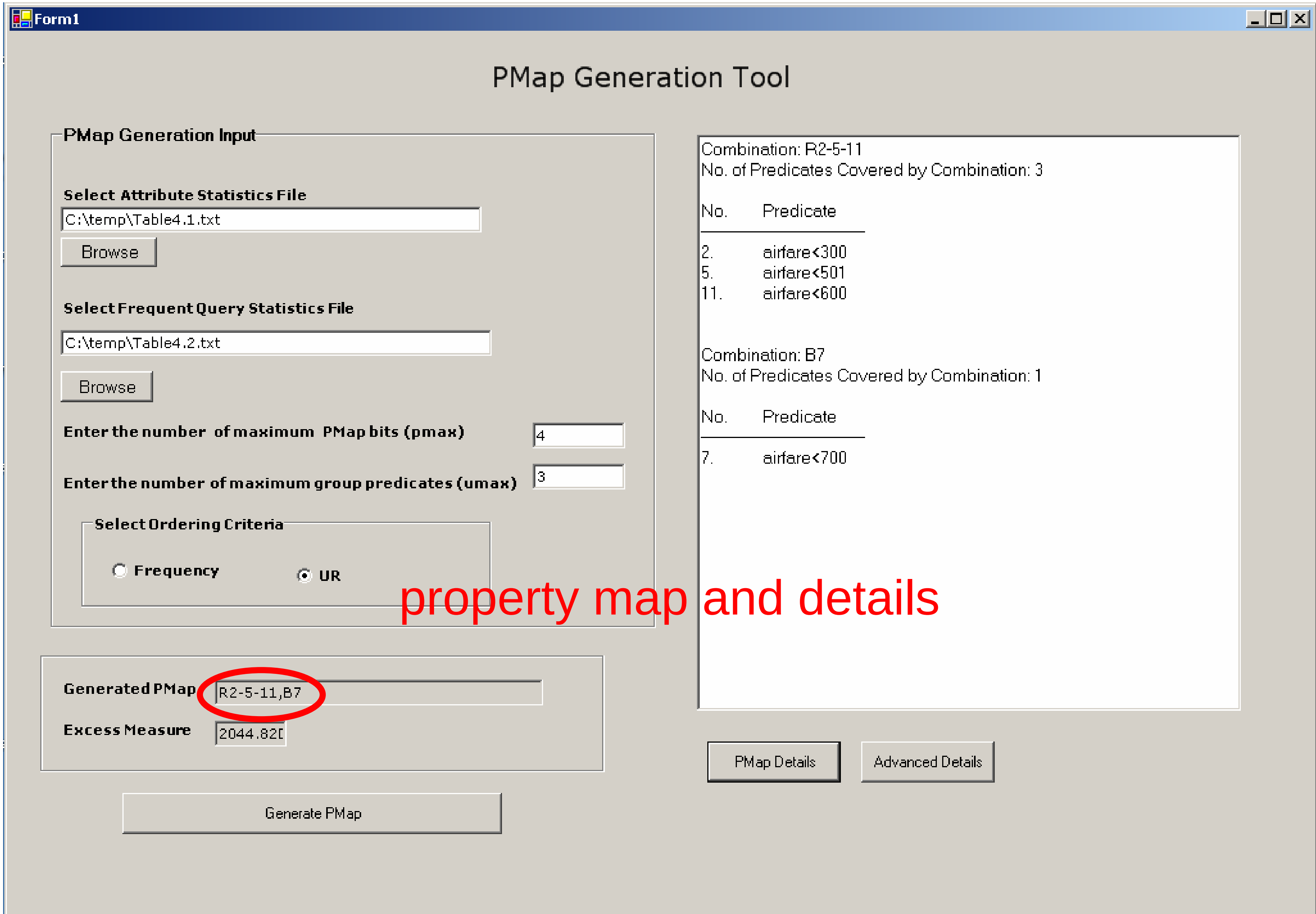
- order properties within a PMap in ascending order of *E* values

Property	Predicate	Bits	Property type	Value set
R2-5-11	airfare < 300 airfare < 501 airfare < 600	2	range	00 = (0-300), 01 = (301-501), 10 = (502,600), 11 = (601-5000)
B7	airfare < 700	1	Boolean	0,1
B1	alcode < 100	1	Boolean	0,1

Screen Shot: Input



Screen Shot: Output



property map and details

Design Algorithms: Evaluation

- compared initial algorithms to enhanced algorithms
- 4 types of queries chosen from BENCH:
  - high selectivity
  - low selectivity
  - high cardinality
  - low cardinality
- workloads formed as 55% of main group, 15% of each of the others: *HS, LS, HC, LC*

Work-load	High selectivity query group	Low selectivity query group	High cardinality query group	Low cardinality query group	Total
HS	99%	77%	74%	62%	86%
LS	99%	95%	75%	65%	85%
HC	99%	78%	87%	59%	84%
LC	99%	94%	76%	80%	82%

Next Steps

- join queries
- integration into a data warehouse system
- realistic evaluation