Quick and Automatic Selection of POMDP Implementations on Mobile Platform Based on Battery Consumption Estimation

Xiao Yang

University of Waterloo

Supervisor:
Jesse Hoey

March 14, 2014

Overview

1. Problem Statement
   - Select POMDP implementations for mobile platforms
2. Solution
   - Selection based on battery depletion rate
   - Mobile battery consumption model
3. Experiment and Evaluation
   - Experimental Settings
   - Results
   - Evaluation
4. Conclusions and Future Plan

Partially Observable Markov Decision Process

Markov Decision Process (MDP)

- agent makes sequential decisions on discrete time
- $< S, A, T(s^{t+1}, s^t, a^t), R(s^{t+1}, s^t, a^t) >$
- maximize long term rewards
- policy - mapping from state to action

Partially Observable MDP (POMDP)

- extended from MDP, involves observations
- maximize long term rewards based on history (actions and observations)
- computationally harder

Combine POMDP with mobile technology

POMDP usually deployed on powerful machines

Why put on mobile devices

- advanced mobile tech provides possibility
- more input from mobile sensors
- increase POMDP's availability

Difficulties - limited resources (CPU, Memory, Battery life, etc.)

Improvements

- work on theory, algorithm
- from the engineering angle

From the eyes of an engineer

Eventually, implemented on mobile devices (different algorithms, mobile-alone or client/server)

Don't do guideline

- too vague to formalize or follow
- existing implementation packages

Select the best from a given set of POMDP execution implementations for a particular problem (it depends)

Selection Criteria

For the good of mobile user experience, what is the most concerning?

Concerns

- slow
- weak signal strength
- crashing down
- short battery life (top gripe)

Focus on battery life

- battery life is visible (opposed to CPU, Memory)
- simplified for research purpose
- battery depletion rate (ranking of implementations as suggestion)

How to obtain battery depletion rate

Straightforward experiments

- record battery depletion rate for every implementation candidates
- low-efficient

Experiments & Estimation

- do some benchmark experiments
- estimate a battery depletion rate based on benchmark and other information

Mobile battery consumption model

Battery consumption comes from different components

- signal standby
- screen display
- CPU
- WIFI/3G
- sensors

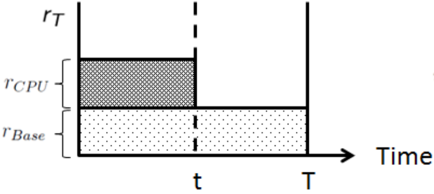
The usage of these components can be translated into battery consumption

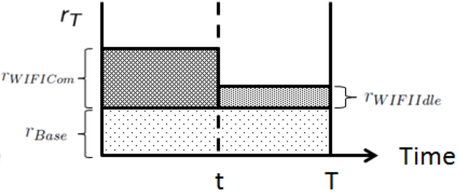
We consider CPU and WIFI (ignore other three)

Mobile battery consumption model details

Suppose the POMDP makes decision every T (actual execution time is t, and the rest $T - t$ is idle)

- $r_T = r_{CPU} * t/T + r_{Base}(T)$ (mobile only, WIFI off)
- $r_T = r_{WIFICom} * t/T + r_{WIFIIdle} * (T - t)/T + r_{Base}(T)$ (mobile/server)





Mobile battery consumption model details

Benchmark: $< r_{CPU}, r_{WIFICom}, r_{WIFIIdle}, r_{Base}(T) >$

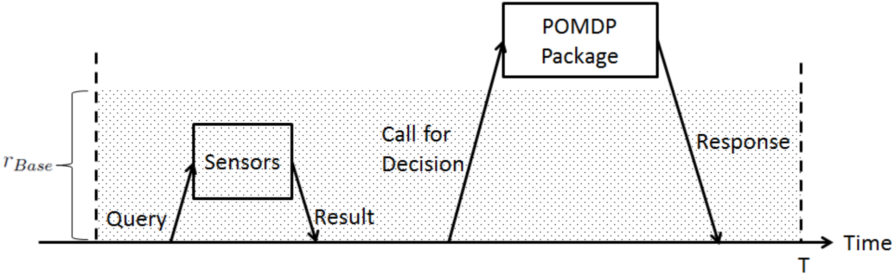
r_{Zero}: BDR of mobile device doing nothing

r_{CPU}: BDR of full cycle CPU usage minus r_{Zero}

$r_{WIFICom}$: BDR of continuous WIFI communication minus r_{Zero}

$r_{WIFIIdle}$: BDR of WIFI on minus r_{Zero}

$r_{Base}(T)$:



Mobile battery consumption model example

For a particular POMDP problem, given a set of implementation candidates implm1, implm2, implm3, ..., we examine the average BDR on three typical time intervals T1, T2, T3:

- obtain benchmark $< r_{CPU}, r_{WIFICom}, r_{WIFIIdle}, r_{Base}(T) >$
- record average actual execution time t for each implementation candidate
- apply to the formula above, get r_{T_1}, r_{T_2}, r_{T_3} for each implementation candidate
- rank implementations based on their $\frac{r_{T_1}+r_{T_2}+r_{T_3}}{3}$

Mobile battery consumption model advantages

Using mobile battery consumption model to estimate BDR is fast

- a lot of implementation candidates to rank, only minutes for each (opposed to actual battery experiment for each)
- benchmark is device-determined, independent from POMDP problem
- benchmark might be shared among similar kind of devices (have no/little effects on ranking)

Experiments & Evaluation Overview

What do we have?

- An mobile device
- A POMDP problem
- Some POMDP implementation packages
- A mobile software architecture that allows POMDP implementations plugin

What do we want to do?

- Compare results from straightforward battery experiments and estimation

What do we expect?

- Rankings are the same from straightforward battery experiments and estimation
- Battery depletion rates from those two are close

Experiments setup details

Mobile device

- Nexus 4, Android 4.2

POMDP problem

- Location and Context Aware Safety Assistant: 2880 states, 72 observations, and 6 actions

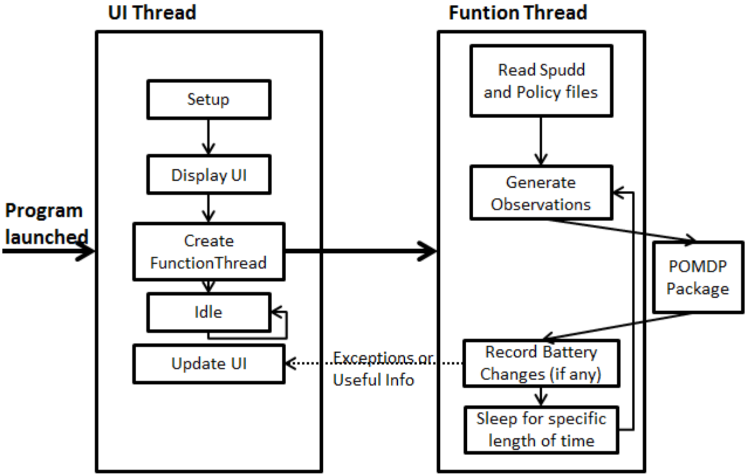
POMDP implementation packages

- FSC, 321 internal states
- Symbolic Perseus, policy calculated by perseus method, 26 α-vectors
- Client/Server, Mobile device runs as client to query on server, POMDP executes on server
- Flat Policy, policy calculated by enumeration algorithm, 26 α-vectors

Experiments setup details

Mobile software architecture

- No SIM card, WIFI on when necessary, Screen brightness fixed to 50%



Straightforward Battery Experiments

Plugin every POMDP implementation candidate, and have each of them run continuously for about 3 hours (record time stamp of every battery level change), with time intervals 10 sec, 2 sec, and 1 sec

Implementation	**T (sec)**	**BDR (%/min)**	**Std Error**
FSC	10	0.159	0.0047
	2	0.154	0.0045
	1	0.175	0.0087
Client/Server	10	0.166	0.0064
	2	0.172	0.0076
	1	0.181	0.0090
Symbolic Perseus	10	0.176	0.0049
	2	0.226	0.0058
	1	0.256	0.0111
Flat Policy	10	0.166	0.0041
	2	0.207	0.0060
	1	0.244	0.0085

Battery Consumption Estimation

Conduct battery experiments to obtain benchmark, and record average per round execution time of each POMDP implementation candidate, apply them to mobile battery consumption mobile to generate estimation

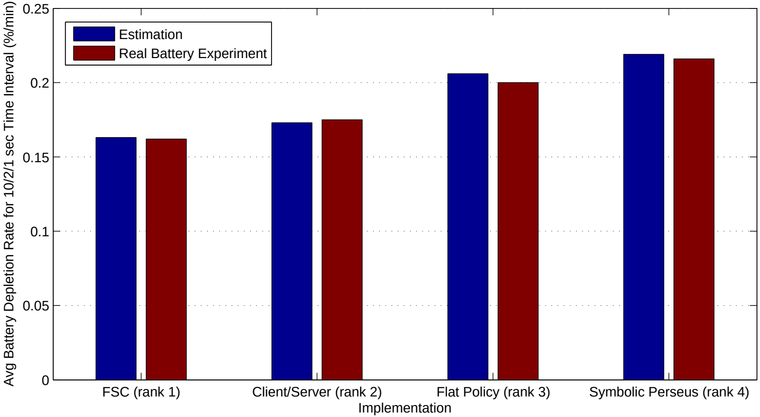
Bnchmrk	T	BDR	StdE
r_{Base}	10	0.158	0.0087
	2	0.157	0.0061
	1	0.172	0.0033
r_{Zero}	N/A	0.148	0.0039
$r_{WithCPU}$	N/A	0.298	0.0132
$r_{WithWIFICom}$	N/A	0.173	0.0071
$r_{WithWIFIIdle}$	N/A	0.149	0.0035

$r_{CPU} = 0.150$
$r_{WIFICom} = 0.025$
$r_{WIFIIdle} = 0.001$

Implementation	T	Avg Exe Time	BDR
FSC	10	0.001 sec	0.158
	2		0.157
	1		0.172
Client/Server	10	0.898 sec	0.166
	2		0.169
	1		0.194
Symbolic Perseus	10	0.669 sec	0.176
	2		0.207
	1		0.272
Flat Policy	10	0.472 sec	0.166
	2		0.192
	1		0.243

Comparison

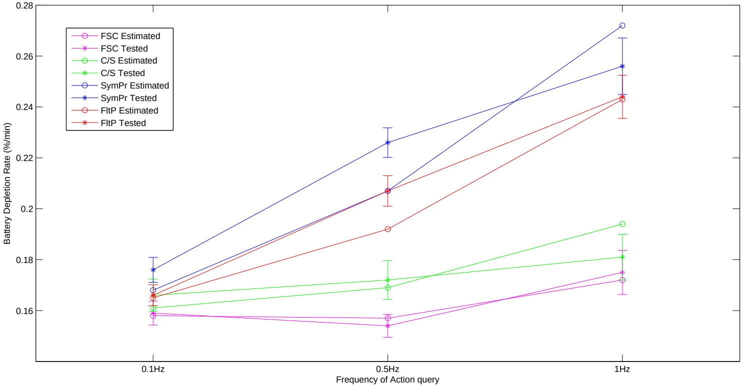
Rankings from straightforward battery experiments and battery estimation are the same



Comparison

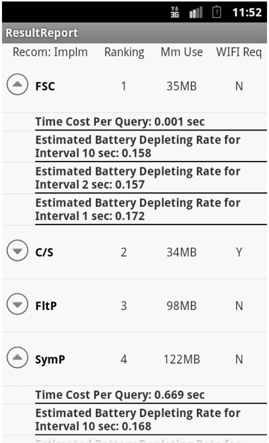
Actual numbers are close, have average cosine-similarity around 0.999

$$similarity(A, B) = cos(A, B) = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^{n} A_i \cdot B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \cdot \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$



Final suggestion for developers

Rank implementations by their battery depletion rates, and list auxiliary information for developers' reference



ResultReport

Recom: Implm	Ranking	Mm Use	WIFI Req
FSC	1	35MB	N

Time Cost Per Query: 0.001 sec

Estimated Battery Depleting Rate for Interval 10 sec: 0.158

Estimated Battery Depleting Rate for Interval 2 sec: 0.157

Estimated Battery Depleting Rate for Interval 1 sec: 0.172

Recom: Implm	Ranking	Mm Use	WIFI Req
C/S	2	34MB	Y
FltP	3	98MB	N
SymP	4	122MB	N

Time Cost Per Query: 0.669 sec

Estimated Battery Depleting Rate for Interval 10 sec: 0.168

Conclusions

- We tackle the POMDP's deployment on mobile from engineering perspectives, focus on POMDP implementation selection based on experiments
- We design and justify the criterion of selecting POMDP implementations - mobile battery consumption
- We propose a method to quickly estimate battery consumption in the long run for POMDP on mobile
- We build a framework to automatically conduct battery experiments on mobile devices for POMDP problems
- Beyond these

Future plans

- More experiments to support the assumption that benchmark can be shared among similar devices
- More experiments on different devices and POMDP problems to verify the our mobile battery consumption model
- Generalize our work to suit for other AI or ML models

Acknowledgment

Jesse Hoey

Pascal Poupart and Dan Lizotte

Marek Grzes

Chengbo Li and Luyuan Lin

The end

Thank you!

- Questions?
- Comments?