A Compromise Programming Approach to Multiobjective Markov Decision Processes

Wlodzimierz Ogryczak[1], Patrice Perny and Paul Weng

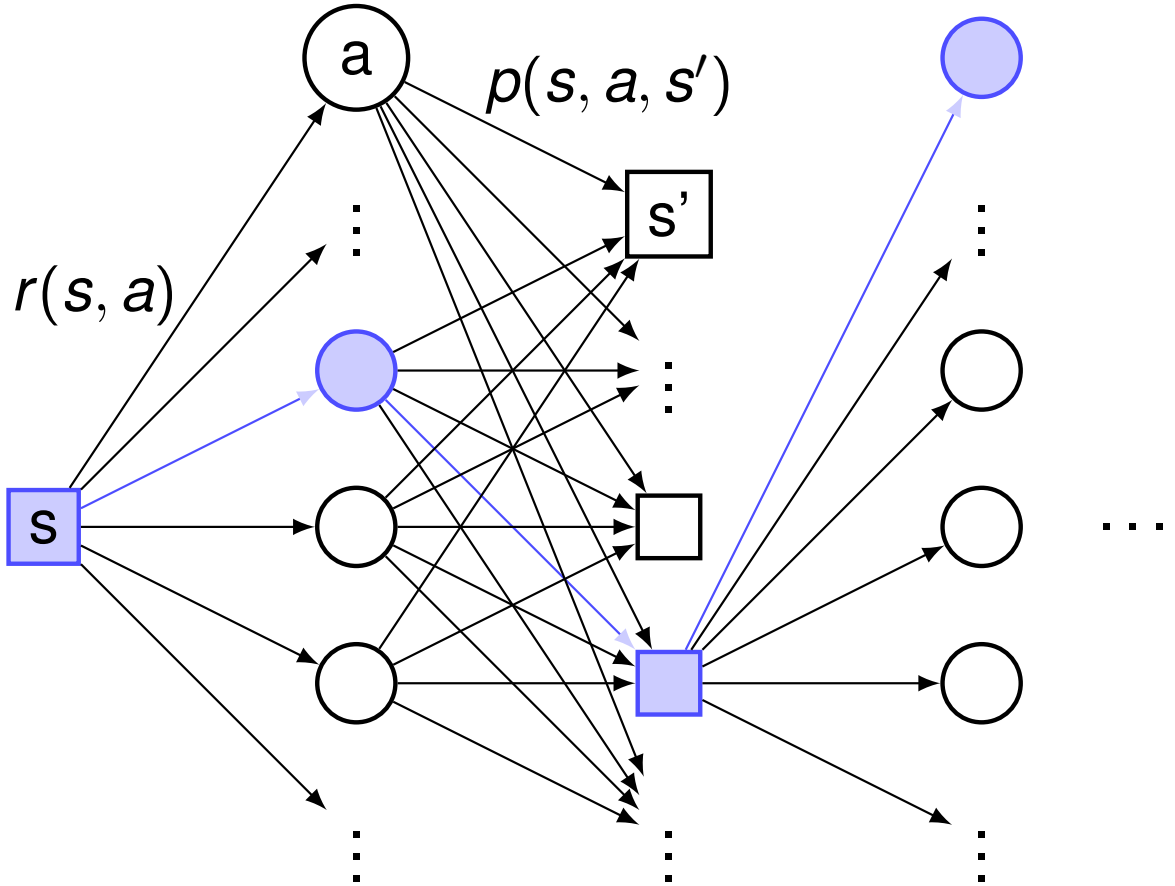
LIP6 - UPMC, Paris, France

June, 13-17 2011

21st International Conference on MCDM
Jyvaskyla, Finland

[1] *on leave from Warsaw University of Technology, Poland*

Sequential Decision Making under Uncertainty



Markov Decision Processes (MDPs)

Definition

- S set of states
- A set of actions
- $p : S \times A \times S \to [0, 1]$
- $r : S \times A \to \mathbb{R}$

Solution

- Pure/randomized decision rule δ
- (Stationary) pure policy π

Value Functions and Solution Methods

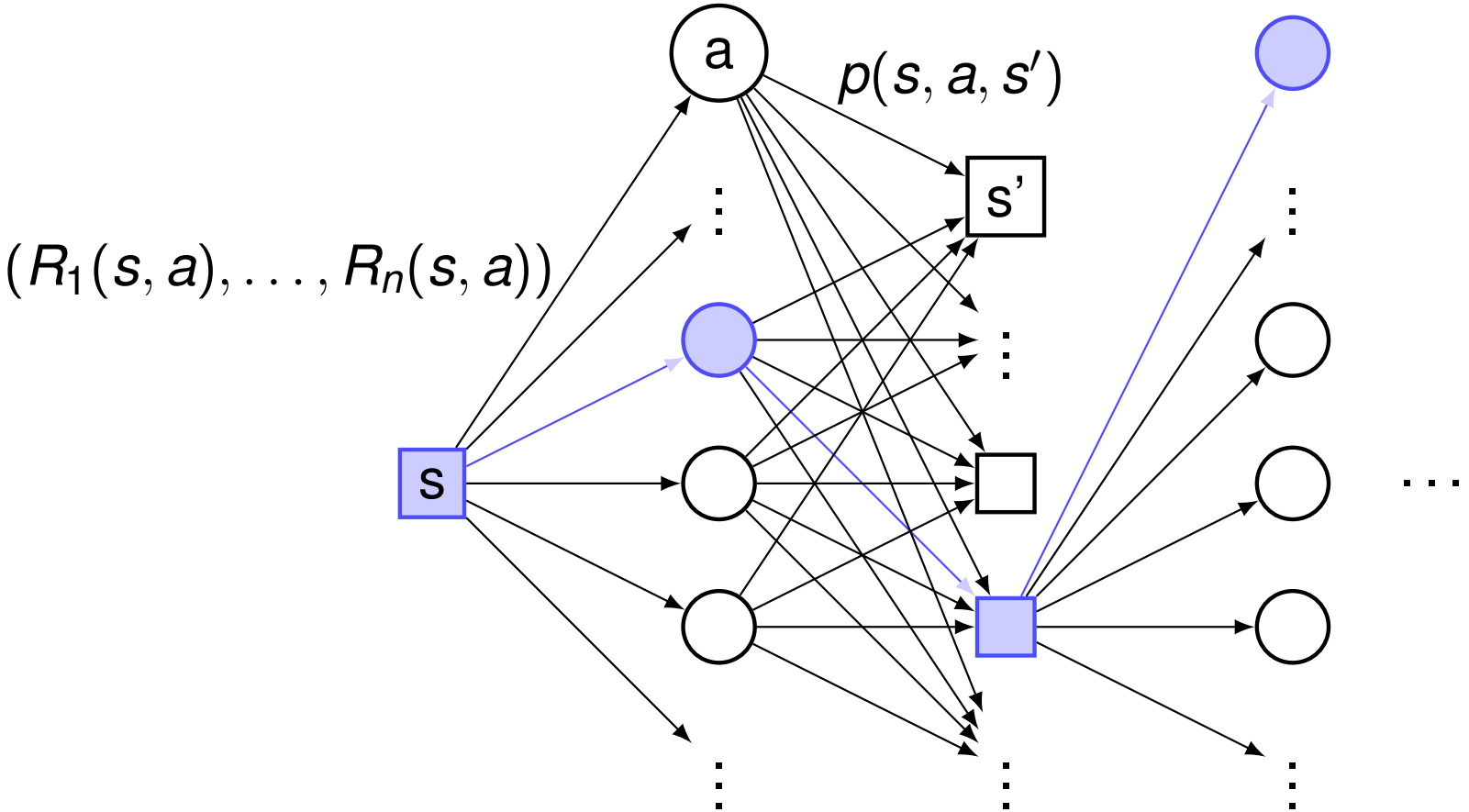
Value functions

- $v_t^\pi(s) = r(s, \delta_t(s)) + \gamma \sum_{s' \in S} p(s, \delta_t(s), s') v_{t-1}^\pi(s')$
- $\pi \gtrsim \pi' \Leftrightarrow \forall s, v^\pi(s) \geq v^{\pi'}(s)$
- $v^*(s) = \max_{a \in A} r(s, a) + \gamma \sum_{s' \in S} p(s, a, s') v^*(s')$

Family of solution methods

- Value/Policy iterations
- LP

Multiobjective MDPs (MMDPs)



Multiobjective MDPs (MMDPs)

Definition

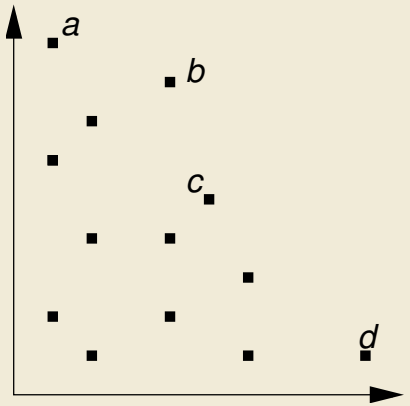
- $R\colon S \times A \to \mathbb{R}^n$ (n criteria)
- $V^\pi(s) \in \mathbb{R}^n$

Value functions

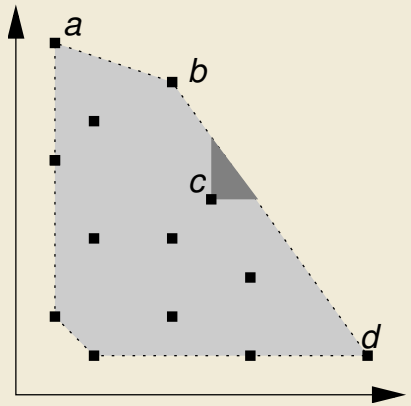
- $V_t^\pi(s) = R(s, \delta_t(s)) + \gamma \sum_{s' \in S} p(s, \delta_t(s), s') V_{t-1}^\pi(s')$
- $\pi \succsim \pi' \Leftrightarrow \forall s,\ V^\pi(s) \geq_P V^{\pi'}(s)$

Scalarizing Function for Compromise Search

Example



Pure policies



Randomized policies

Scalarizing Function ψ

- $\psi : \mathbb{R}^n \to \mathbb{R}$ monotonic w.r.t. Pareto dominance
- $v(s) = \psi(V_1(s), \ldots, V_n(s))$

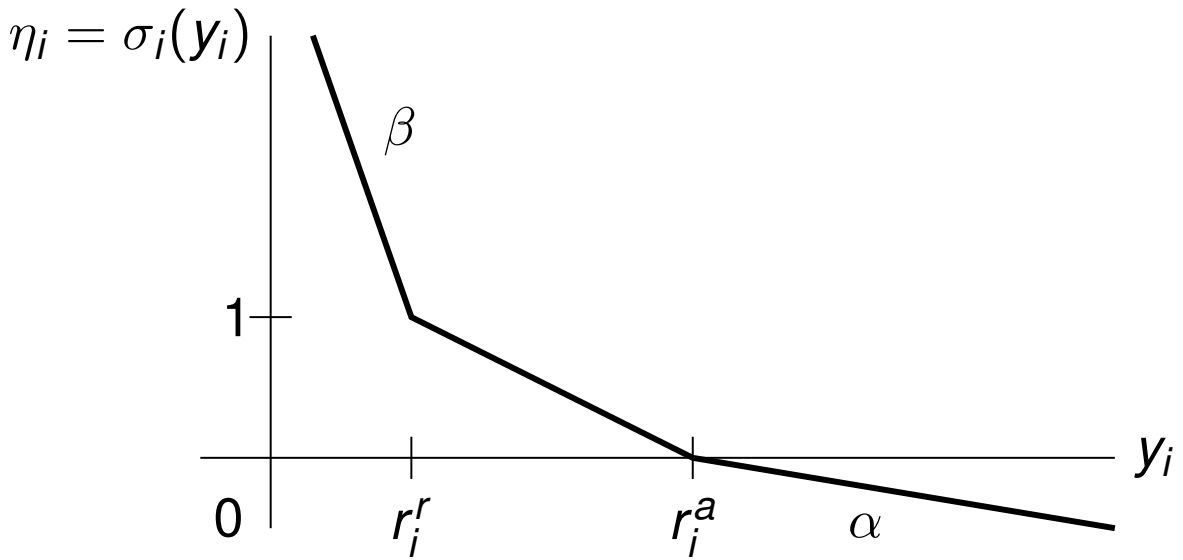
- Weighted sum does not provide any control on tradeoffs

Reference Point Method (RPM)

Generic Scalarizing Achievement Function (Wierzbicki, 82)

$$\psi_\varepsilon(y) = (1-\varepsilon)\max_{i=1\ldots n}\sigma_i(y_i) + \frac{\varepsilon}{n}\sum_{i=1\ldots n}\sigma_i(y_i)$$

$$\sigma_i(y_i) = \frac{1}{r_i^r - r_i^a}\max\left\{\beta y_i + (1-\beta)r_i^r - r_i^a, y_i - r_i^a, \alpha(y_i - r_i^a)\right\}$$



RPM with an OWA

OWA

$$OWA(\eta) = \sum_{i=1}^{n} \omega_i \eta_{\langle i \rangle} \quad \text{where} \quad \eta_i = \sigma_i(y_i) \quad \forall\, i = 1 \ldots n$$

where $\omega_1 > \omega_2 > \ldots > \omega_n > 0$ and $\eta_{\langle 1 \rangle} \geq \eta_{\langle 2 \rangle} \geq \ldots \geq \eta_{\langle n \rangle}$

Example

$r^r = (0,0,0) \quad r^a = (10,10,10) \quad w = (5/10, 3/10, 2/10)$

y	η	$\eta_{\langle 1 \rangle}$	$\eta_{\langle 2 \rangle}$	$\eta_{\langle 3 \rangle}$	OWA	ψ_0	ψ_ε
$(4,5,9)$	$(6,5,1)$	6	5	1	4.7	6	$6+4\varepsilon$
$(4,8,6)$	$(6,2,4)$	6	4	2	4.6	6	$6+4\varepsilon$
$(4,7,7)$	$(6,3,3)$	6	3	3	4.5	6	$6+4\varepsilon$

Main properties of OWA

- Symmetry:

$$OWA(\eta_1, \ldots, \eta_n) = OWA(\eta_{\tau(1)}, \ldots, \eta_{\tau(n)})$$

- Pareto-Monotonicity:

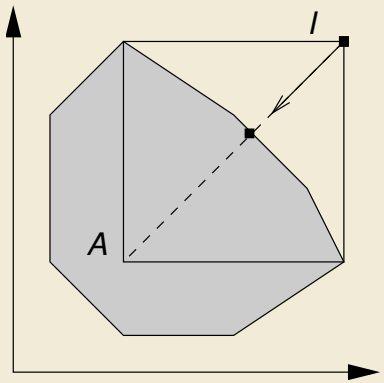
$$\eta \succ_P \eta' \Rightarrow OWA(\eta) > OWA(\eta')$$

- Fairness (Monotonicity w.r.t Pigou-Dalton transfers):
$\forall i, j \in \{1, \ldots, n\}$ s.t $\eta_i > \eta_j, \forall \varepsilon \in (0, \eta_i - \eta_j)$,

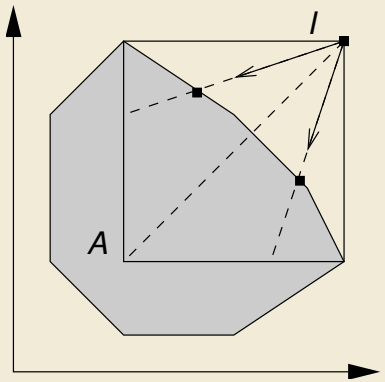
$$OWA(\eta_1, \ldots, \eta_i - \varepsilon, \ldots, \eta_j + \varepsilon, \ldots, \eta_n) < OWA(\eta_1, \ldots, \eta_n)$$

RPM with a Weighted OWA

OWA symmetric on regrets ⇒ WOWA



Compromise Solution



Different Weights

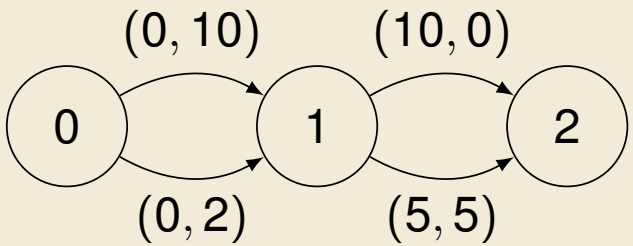
RPM WOWA

$$WOWA(\eta) = \sum_{i=1}^{n} w_i(\lambda, \eta)\eta_{\langle i \rangle}$$

where $w_i(\lambda, \eta) = \varphi(\sum_{k \leq i} \lambda_{\tau(k)}) - \varphi(\sum_{k<i} \lambda_{\tau(k)})$ with φ a monotone increasing function that interpolates points $(\frac{i}{m}, \sum_{k \leq i} \omega_k)$ together with the point (0.0).

Difficulty of Compromise Search in MDPs

Example: Not dynamically consistent



- Best Compromise in 1 : Down $\succ$ Up
- Best compromise in 0 : (**Up**, Down) $\prec$ (**Up**, Up)

- Dependent on the initial state
- $\pi \succ \pi' \nRightarrow (\delta, \pi) \succ (\delta, \pi')$

$\Rightarrow$ Optimal policies cannot be obtained from optimal sub-policies (true for Tchebycheff, OWA, WOWA...)

Solving Standard MDPs by LP

Primal LP

$$\begin{aligned} \min \quad & \sum_{s \in S} \mu(s) v(s) \\ \text{s.t.} \quad & v(s) \geq r(s,a) + \gamma \sum_{s' \in S} p(s,a,s') v(s') \quad \forall s \in S, \forall a \in A \end{aligned}$$

Dual LP

$$\max \quad \sum_{s \in S} \sum_{a \in A} r(s,a) x(s,a)$$

$$\text{s.t.} \quad \begin{cases} \sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a) p(s',a,s) = \mu(s) \quad \forall s \in S \\ x(s,a) \geq 0 \quad \forall s \in S, \forall a \in A \end{cases}$$

MMDP as MOLP

(Viswanathan et al., 77)

$$\max \quad y_i = \sum_{s \in S} \sum_{a \in A} R_i(s,a) x(s,a) \quad \forall\, i = 1, \ldots, n$$

$$\text{s.t.} \quad \begin{cases} \sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a) p(s',a,s) = \mu(s) \quad \forall s \in S \\ x(s,a) \geq 0 \quad \forall s \in S, \forall a \in A \end{cases}$$

Solving Method (1/2)

NLP formulation

$$\min \quad WOWA(\eta)$$

$$\text{s.t.} \begin{cases} \eta_i = \max\{\beta \dfrac{y_i - r_i^r}{r_i^r - r_i^a} + 1,\ \dfrac{y_i - r_i^a}{r_i^r - r_i^a},\ \alpha \dfrac{y_i - r_i^a}{r_i^r - r_i^a}\} & \forall\, i = 1 \dots n \\ y_i = \displaystyle\sum_{s \in S} \sum_{a \in A} R_i(s,a) x(s,a) & \forall\, i = 1 \dots n \\ \displaystyle\sum_{a \in A} x(s,a) - \gamma \sum_{s' \in S} \sum_{a \in A} x(s',a) p(s',a,s) = \mu(s) & \forall s \in S \\ x(s,a) \geq 0 & \forall s \in S, \forall a \in A \end{cases}$$

Solving Method (2/2)

LP reformulation

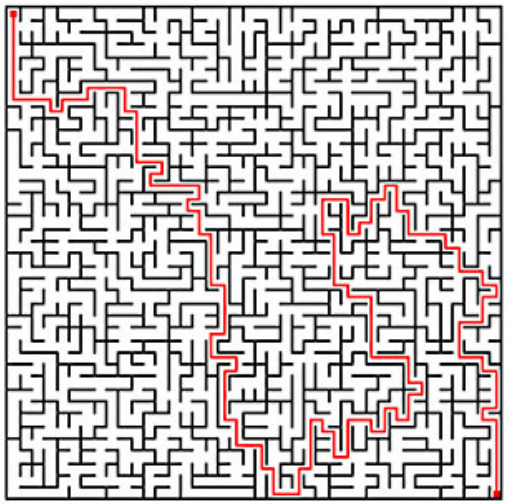
$$\min \quad \sum_{k=1}^{n} \omega'_k z_k$$

$$\text{s.t.} \quad \begin{cases} z_k = kt_k + n \sum_{i=1\ldots n} \lambda_i d_{ik} & \forall\, k = 1 \ldots n \\ \eta_i \leq t_k + d_{ik},\ d_{ik} \geq 0 & \forall\, i = 1 \ldots n, \forall\, k = 1 \ldots n \\ \eta_i \geq \beta(y_i - r_i^r)/(r_i^r - r_i^a) + 1 & \forall\, i = 1 \ldots n \\ \eta_i \geq (y_i - r_i^a)/(r_i^r - r_i^a) & \forall\, i = 1 \ldots n \\ \eta_i \geq \alpha(y_i - r_i^a)/(r_i^r - r_i^a) & \forall\, i = 1 \ldots n \\ y_i = \sum_{s\in S} \sum_{a\in A} R_i(s,a) x(s,a) & \forall\, i = 1 \ldots n \\ \sum_{a\in A} x(s,a) - \gamma \sum_{s'\in S} \sum_{a\in A} x(s',a) p(s',a,s) = \mu(s) \quad \forall s \in S & \\ x(s,a) \geq 0 \quad \forall s \in S, \forall a \in A & \end{cases}$$

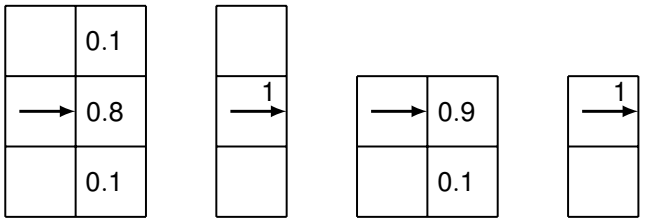
Idea of the Linearization: the Example of OWA

- $OWA(\eta) = \sum_{k=1}^{n} w'_k L_k(\eta)$ $\quad \begin{cases} L_k(\eta) = \sum_{i=1}^{k} \eta_{\langle i \rangle} \\ w'_i = w_i - w_{i+1} \quad \forall\, i = 1 \ldots n-1 \\ w'_n = w_n \end{cases}$
- $L_k(\eta) = \max_{u_{ik}} \{ \sum_{i=1}^{n} \eta_i u_{ik} : \sum_{i=1}^{n} u_{ik} = k, 0 \leq u_{ik} \leq 1, \forall i \}$
- $L_k(\eta) = \min_{t_k, d_{ik}} \{ k t_k + \sum_{i=1}^{n} d_{ik} : \eta_i \leq t_k + d_{ik},\ d_{ik} \geq 0, \forall i \}$
- $\min OWA(\eta) = \begin{cases} \min & \sum_{k=1}^{n} w'_k (k t_k + \sum_{i=1}^{n} d_{ik}) \\ \text{s.t.} & \begin{cases} \eta_i \leq t_k + d_{ik} & \forall i, k \\ d_{ik} \geq 0 & \forall i, k \end{cases} \end{cases}$

Experiments: Navigation in a Grid

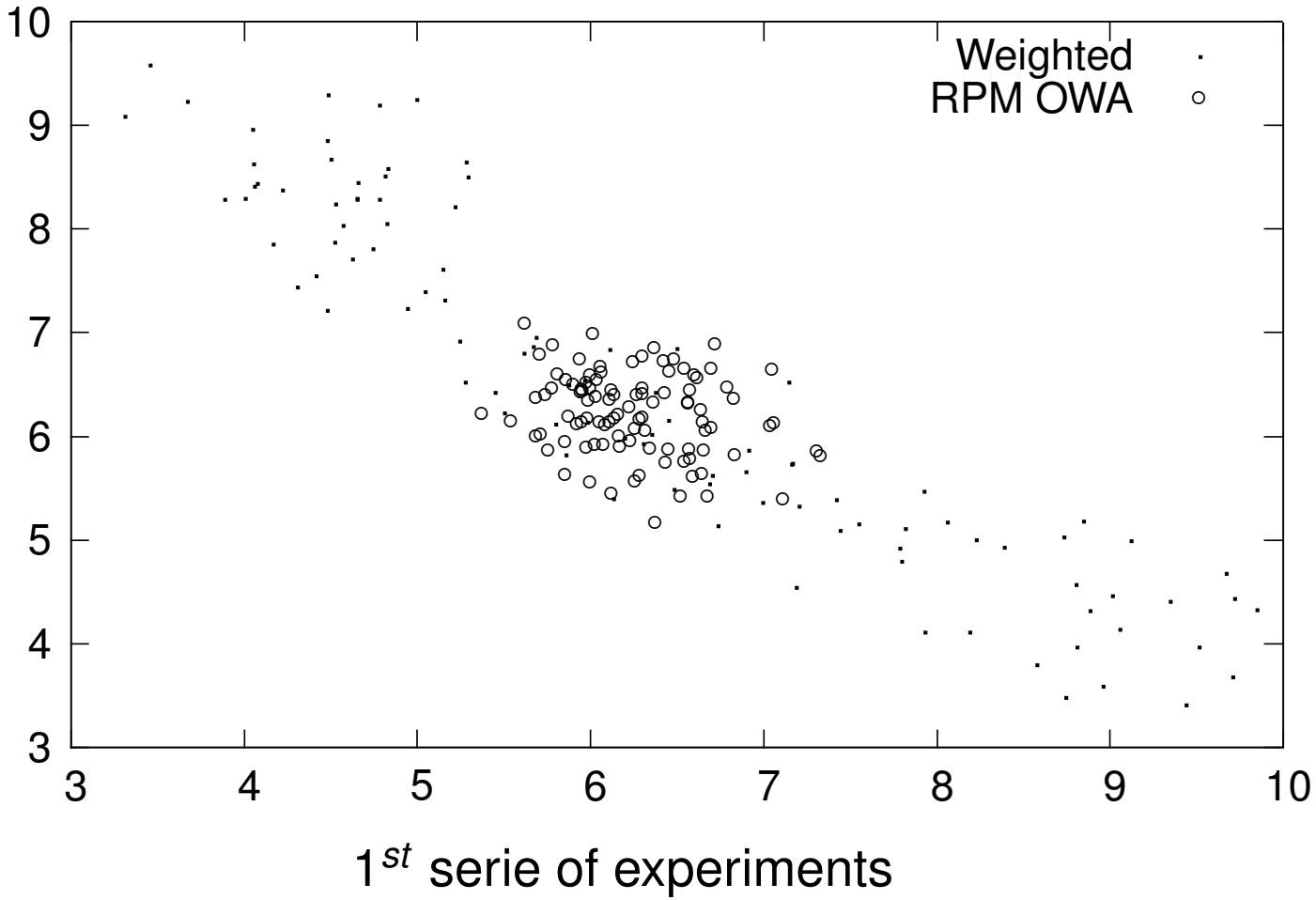


- $S = N \times N$
- $A = \{U, D, L, R\}$
- p
- R randomly drawn in $[0, 1]^2$

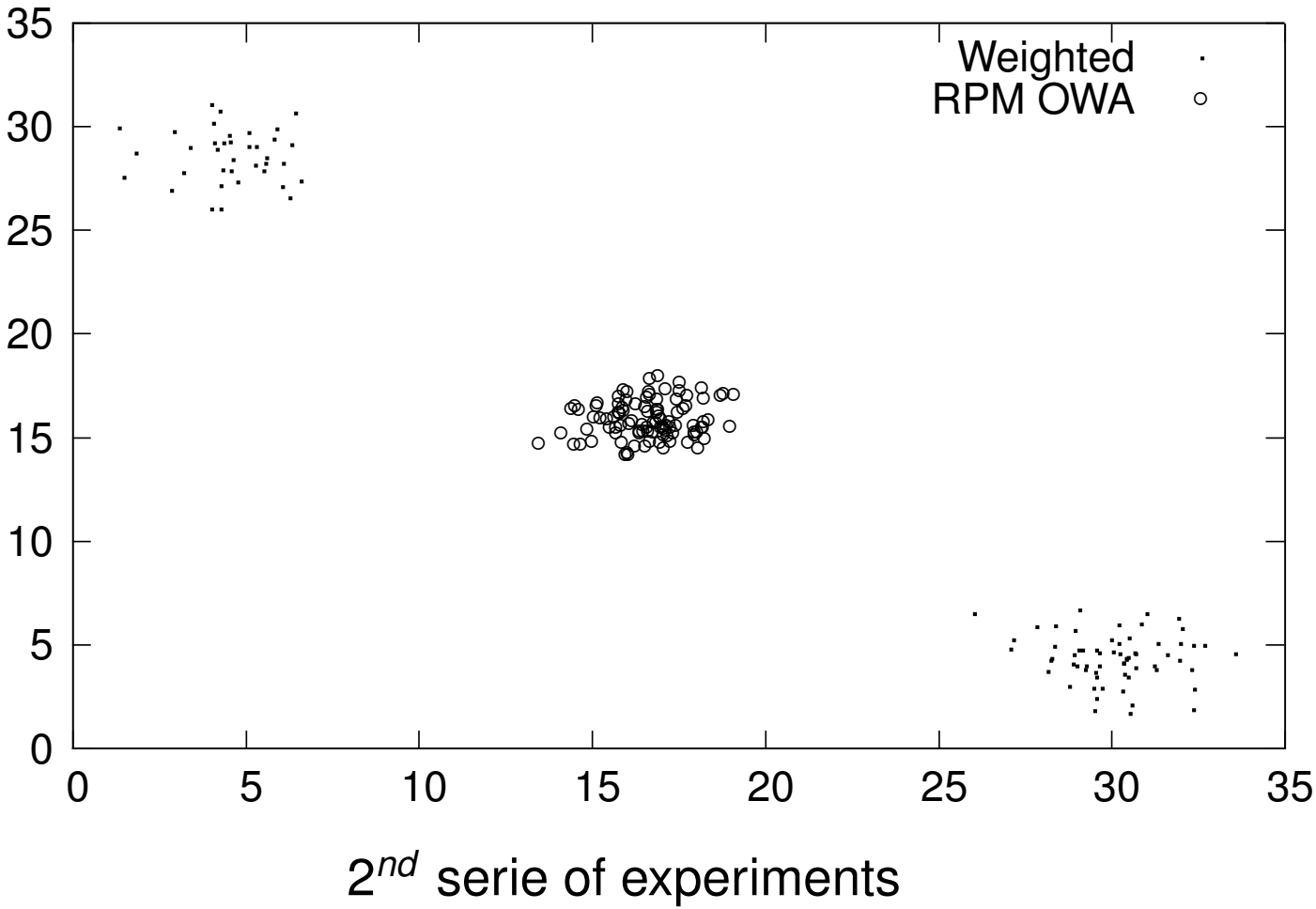


Transitions

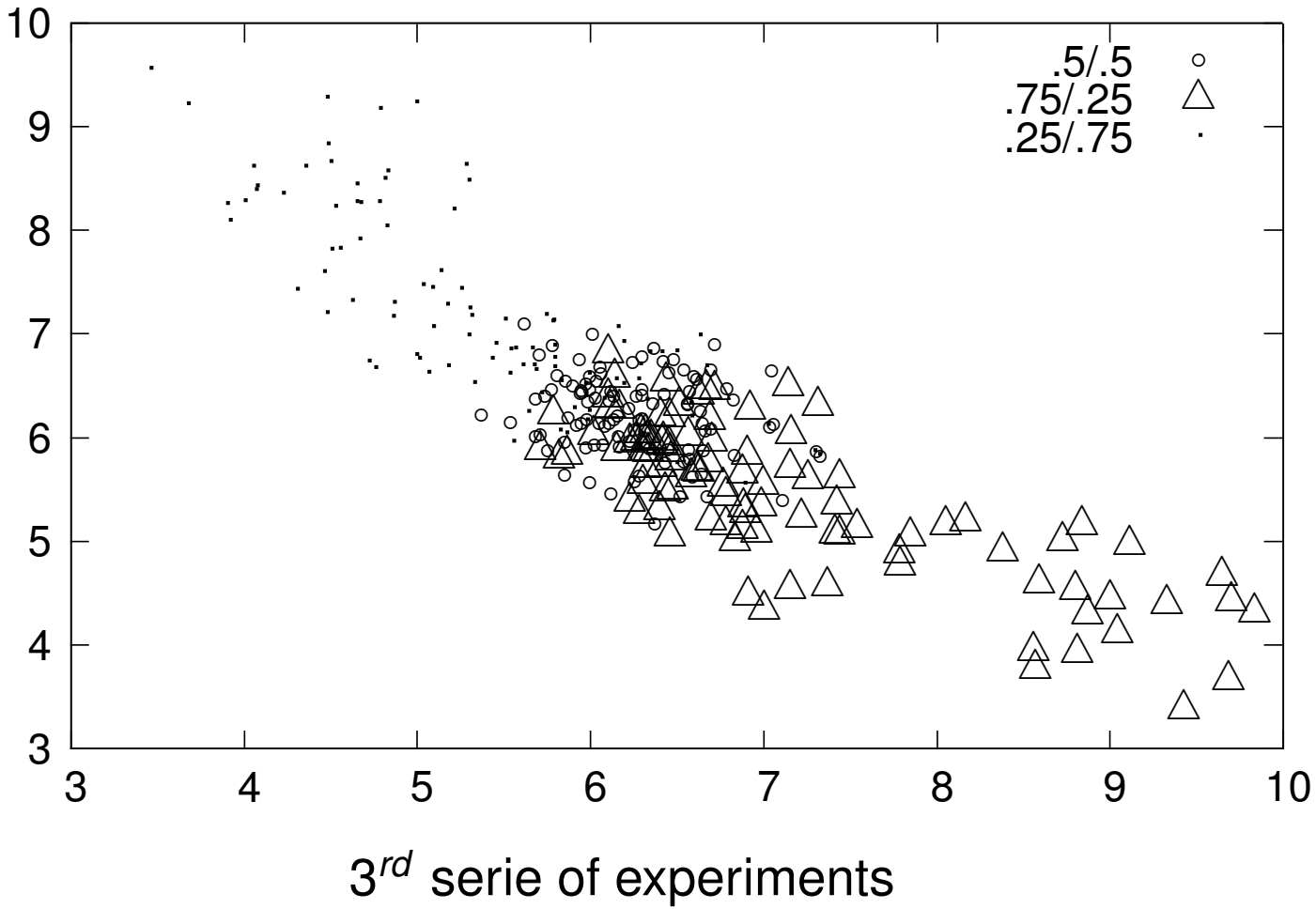
Results on the Navigation Problem (1/3)



Results on the Navigation Problem (2/3)



Results on the Navigation Problem (3/3)



Computation Times

Table: Average execution time in seconds

n	Size	W	RPM OWA	RPM WOWA
2	400	0.17	0.48	0.46
2	2500	5.13	15.06	15.12
2	10000	151.51	417.02	422.06
4	400	0.12	0.75	0.76
4	2500	5.2	28.21	28.27
4	10000	154	821.27	829.83
8	400	0.12	1.3	1.3
8	2500	4.96	50.62	50.72
8	10000	158.26	1514.21	1538.19

Conclusion and Future Work

- Best compromise solution in MMDPs
  - LP approach based on a WOWA scalarizing function
- Other non linear scalarizing functions
  - Choquet integral,...
- Factorized MDP and MMDP
  - to overcome the curse of dimensionality

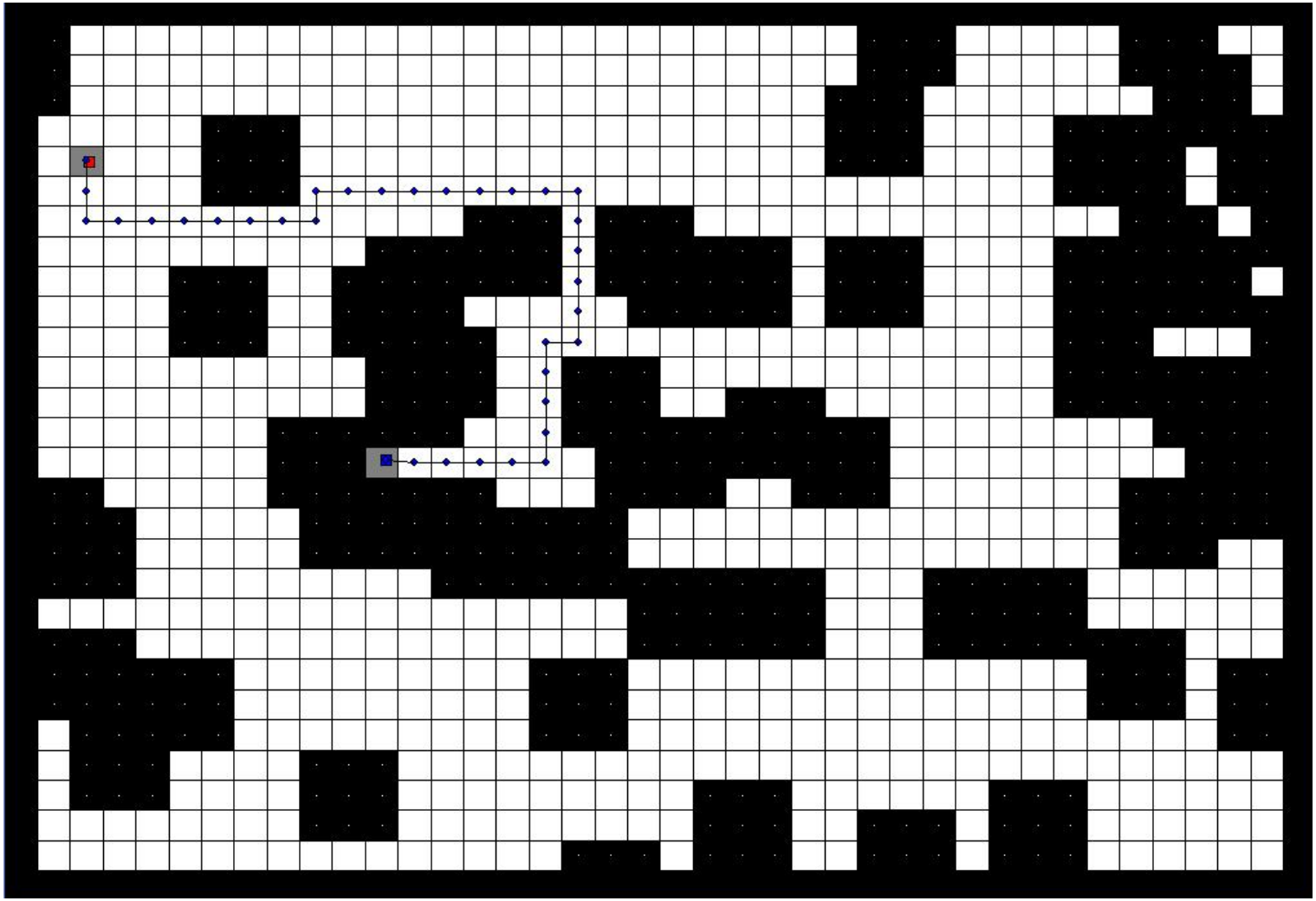
Proposition

For any positive and strictly decreasing normalized preferential weights $\omega_1 > \omega_2 > \ldots > \omega_n > 0$ *and any positive importance weights* λ_i*, if* $\bar{y}$ *is a properly nondominated with tradeoffs bounded by* $\Delta = n\bar{\lambda}\beta\omega_1/(1 - n\bar{\lambda}\omega_1)$ *where* $\bar{\lambda} = \min_{i \in I} \lambda_i$*, i.e. if for any attainable outcome vector* y *the implication*

$$y_i > \bar{y}_i \text{ and } y_k < \bar{y}_k \quad \Rightarrow \quad \frac{y_i - \bar{y}_i}{\bar{y}_k - y_k} \leq \Delta = \frac{n\bar{\lambda}\beta\omega_1}{(1 - n\bar{\lambda}\omega_1)} \qquad (1)$$

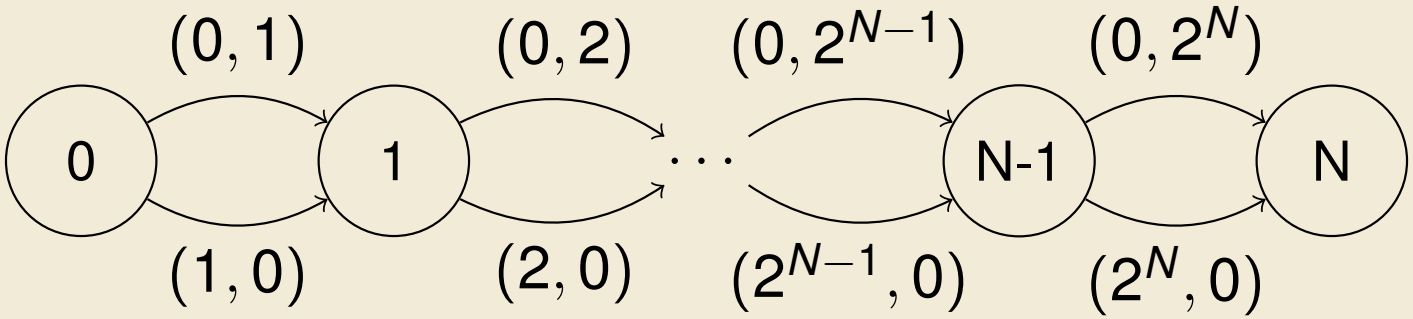
is valid for any $i, k \in I$*, then there exist aspirations levels* r_i^a *and reservation levels* r_i^r *such that* $\bar{y}$ *is an optimal solution of the corresponding problem.*

Example: Path-planning problems

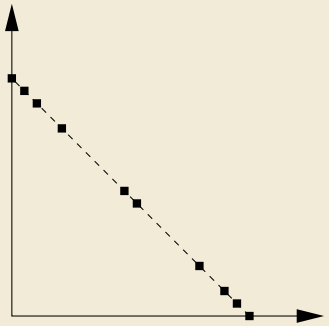


Do we really need all the Pareto optimal solutions?

Example adapted from (Hansen, 80)



$$V_1^{\pi}(0) + V_2^{\pi}(0) = \sum_{i=0}^{N} 2^i = 2^{N+1} - 1$$



- The number of Pareto optimal pure policies grows exponentially with the number of states