LDPC 부호의 Degree Distribution 설계 방법

2013년 통신정보 합동학술대회

10000001000001100001010001111001000101100111010100111110100001110001001001101101011011110110001101001011101110011001010101111111

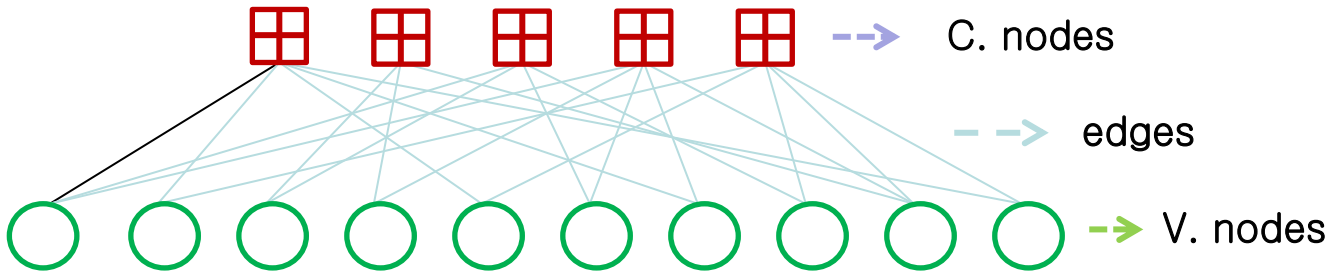
2013.05.02

연세대학교

부호 및 암호 연구실

송민규, 박진수, 송홍엽

LDPC Codes



$$\lambda(x) = \sum_{i=2}^{d_v} \lambda_i x^{i-1} \qquad \rho(x) = \sum_{i=2}^{d_c} \rho_i x^{i-1}$$

$$\text{Code rate } R = \frac{K}{N} = 1 - \frac{\sum_{i=2}^{d_c} \frac{\rho_i}{i}}{\sum_{i=2}^{d_v} \frac{\lambda_i}{i}}$$

λ_i : fraction of edges which are connected to degree i variable nodes
ρ_i : fraction of edges which are connected to degree i check nodes

Ensemble Optimize Problems

There are two objects

- Optimize code rate R for given threshold
- Threshold s^* for given code rate → practically important!

Optimize s^*

Minimize s^*

subject to

$$\sum_{i=1}^{d_c} \rho_i = 1$$
$$\sum_{i=1}^{d_v} \lambda_i = 1$$
$$p_e \to 0 \; as \; l \to \infty$$

by tuning

$$\rho(x), \lambda(x)$$

for given

$$R, d_v, d_c$$

- Richardson 01

Optimize R

Maximize R

subject to

$$\sum_{i=1}^{d_c} \rho_i = 1$$
$$\sum_{i=1}^{d_v} \lambda_i = 1$$
$$p_e \to 0 \; as \; l \to \infty$$

by tuning

$$\rho(x), \lambda(x)$$

for given

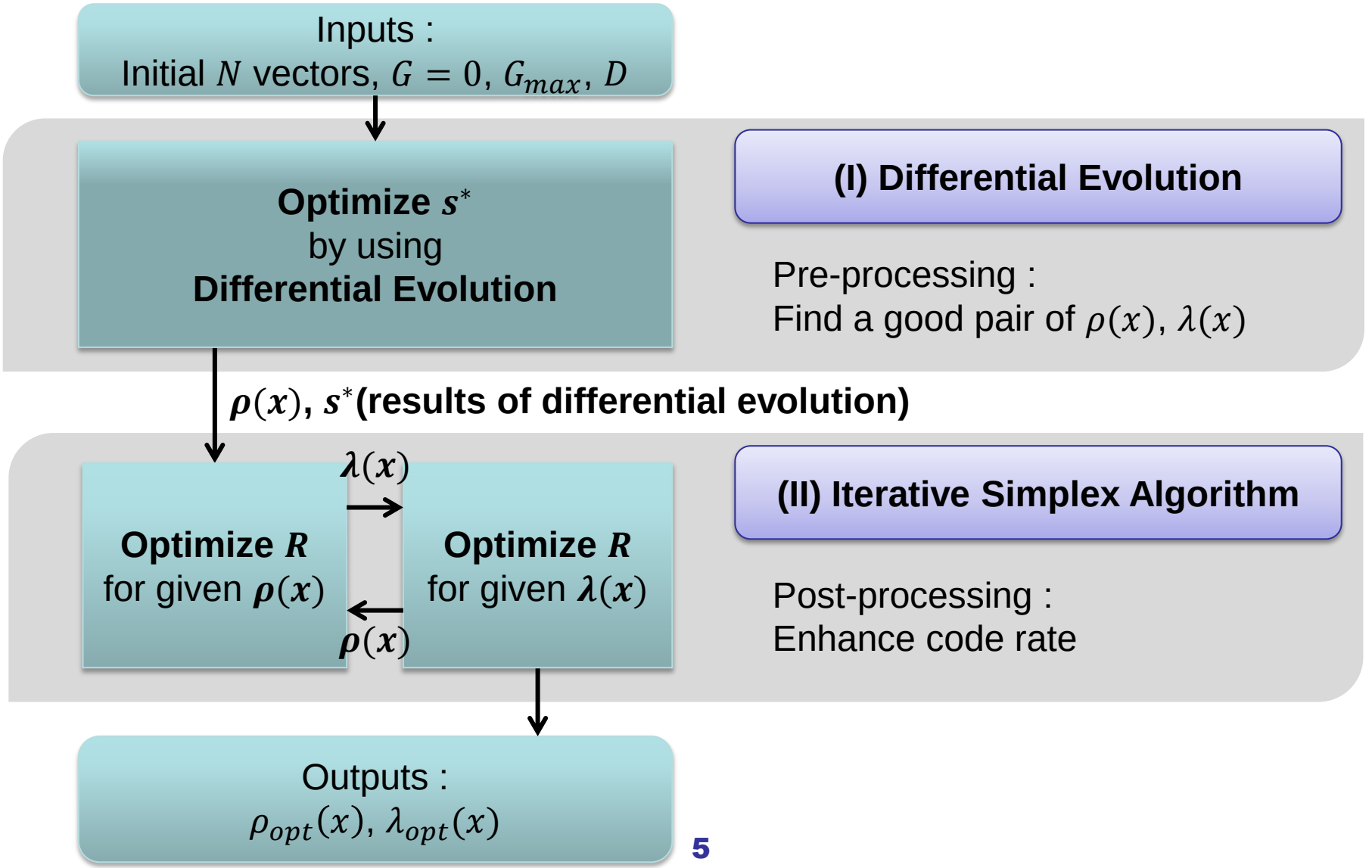
$$s^*, d_v, d_c$$

- Sae-Yong Chung 01

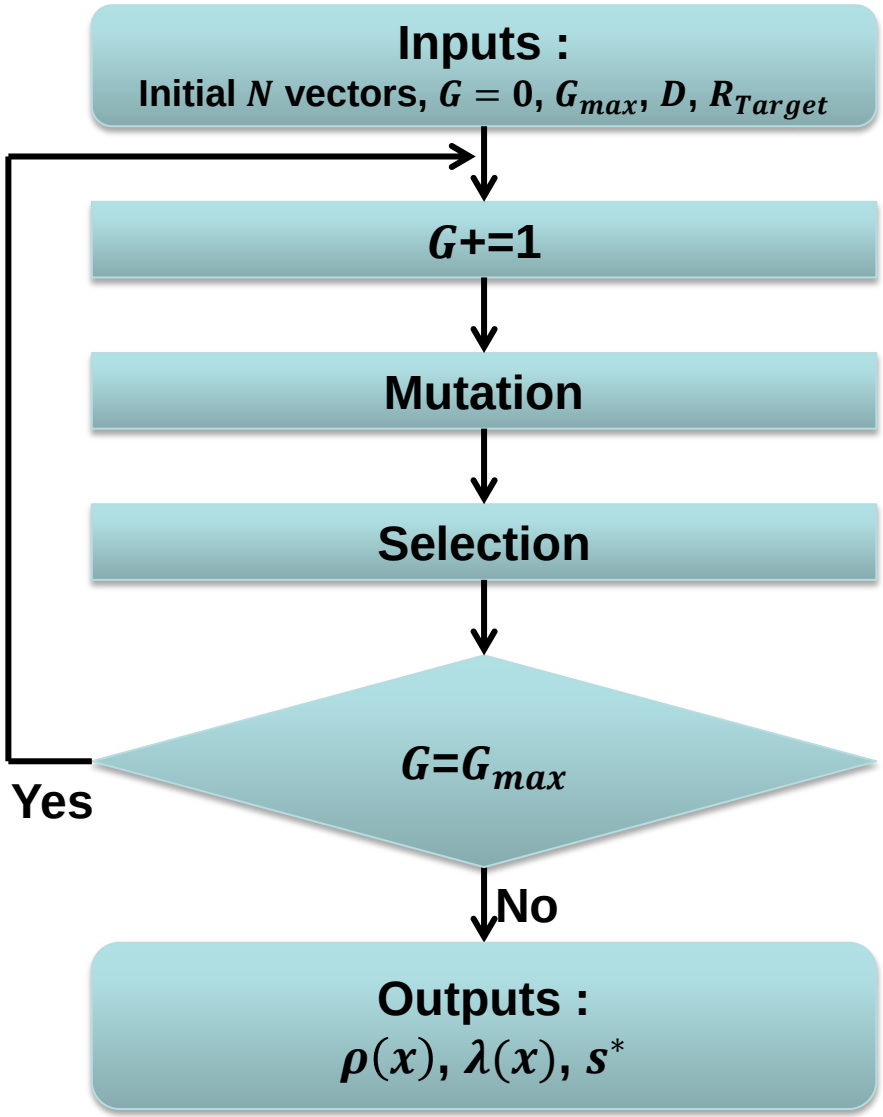
Simplex Method and Differential Evolution

	- George Dantzig 47	- Price 97
	Simplex method	**Differential evolution**
Type	**Optimization algorithm for linear programming (LP) problems**	**Heuristic method (direct parallel search)**
Requirement	**Standard form of LP problems**	**System parameter**
Solution	**Global optimum**	**Close to Global optimum**
Stopping criterion	**Optimality of current basic solution is guaranteed**	**Fixed iteration number**

Main Result : Proposed Scheme



(I) Differential Evolution



Initial N vectors :

$$x_i = \begin{bmatrix} {\rho_{d_c-1,i}}^0 \\ {\rho_{d_c,i}}^0 \\ {\lambda_{1,i}}^0 \\ \vdots \\ {\lambda_{d_v,i}}^0 \end{bmatrix}, 0 \le i \le N$$

Mutation :

$$\boldsymbol{v_i}^{\boldsymbol{G}} = {\boldsymbol{x_{best}}}^{\boldsymbol{G-1}} + \sum_{\boldsymbol{k=1}}^{\boldsymbol{D}} ({\boldsymbol{x_{r1,k}}}^{\boldsymbol{G-1}} - {\boldsymbol{x_{r2,k}}}^{\boldsymbol{G-1}})$$

Controlled Parameters :

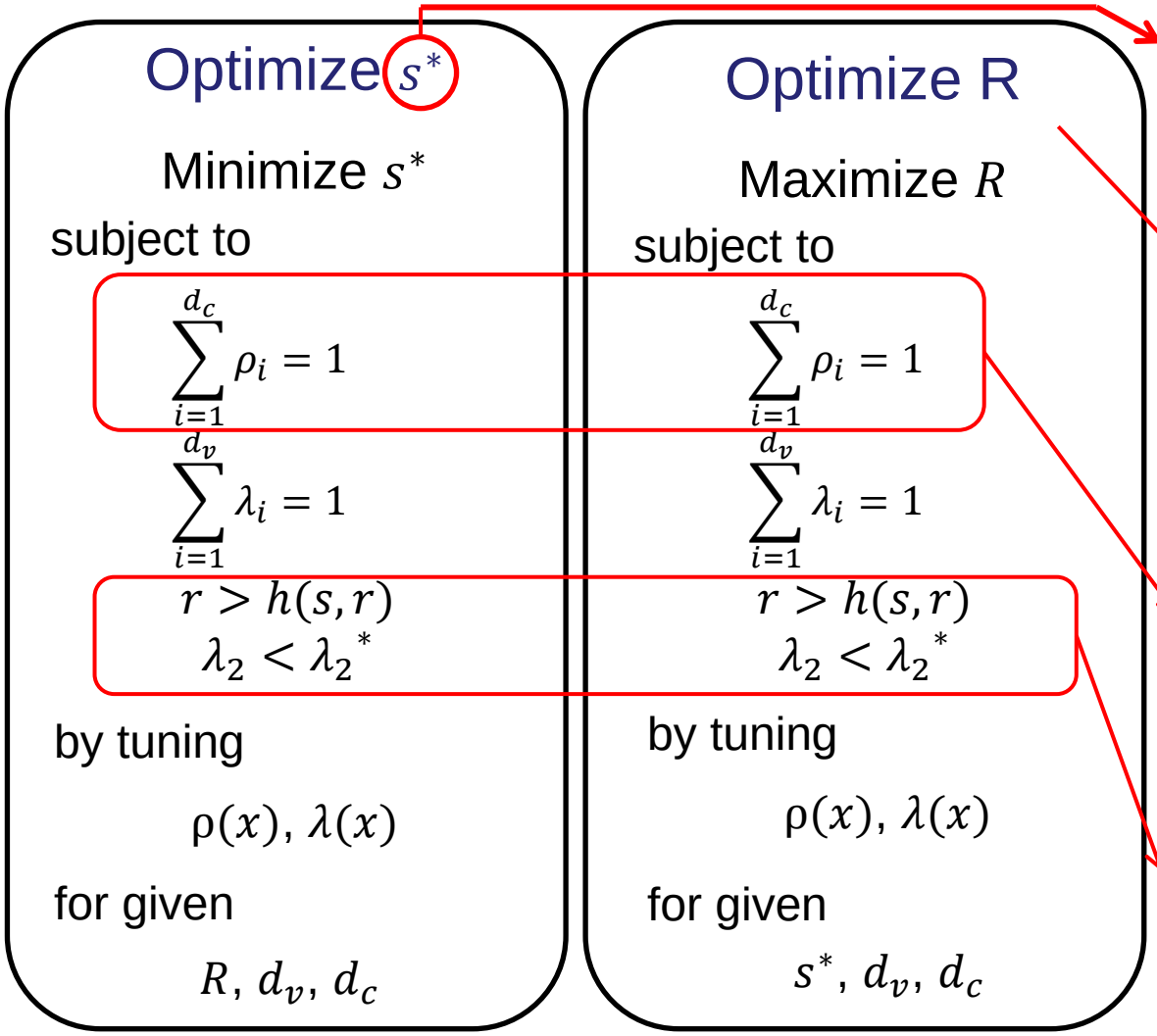
N, G_{max}, D

(I) Differential Evolution : Issues

- Implementation issue
  - $\phi(x)$ and $\phi^{-1}(x)$
  - Size of N
  - Diversity
  - Iteration number
  - Cumulative error
  - Threshold calculation

$$\phi(x) \triangleq \begin{cases} 1 - \frac{1}{\sqrt{4\pi x}} \int_R \tanh\frac{u}{2} e^{-\frac{(u-x)^2}{4x}} du, & if\ x > 0 \\ 1 & if\ x \leq 0 \end{cases},$$

(II) Iterative Simplex Algorithm : Applying LP Solver



The representation of s^* with $\rho(x)$ and $\lambda(x)$ is not known.

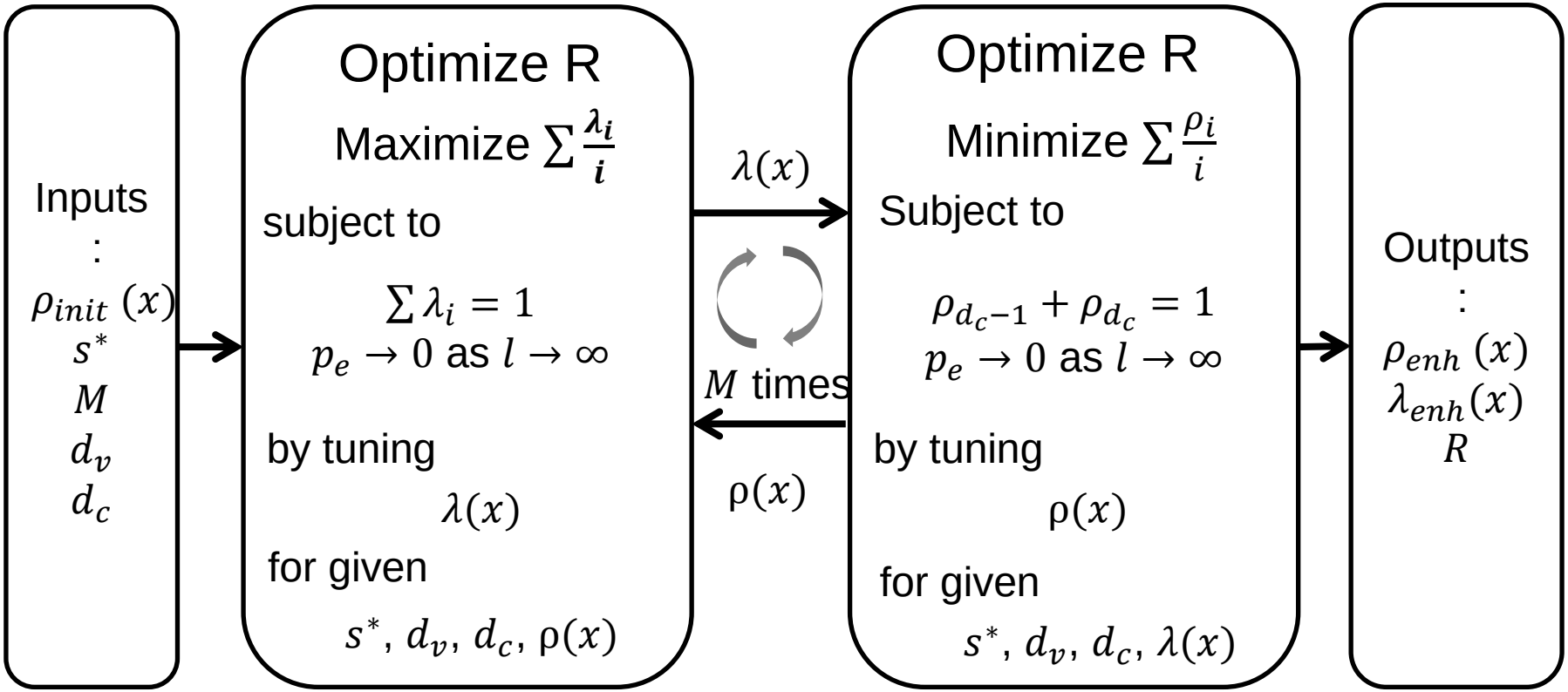
It is not linear program for controlled parameter $\rho(x)$, $\lambda(x)$. But if one of them is given, then **it will be linear program**.

It is known that the $\rho(x) = \rho x^{k-1} + (1-\rho)x^k$ is good enough to obtain good performance.

$p_e \to 0$ condition, by Gaussian approximation.

- Sae-Young Chung 01

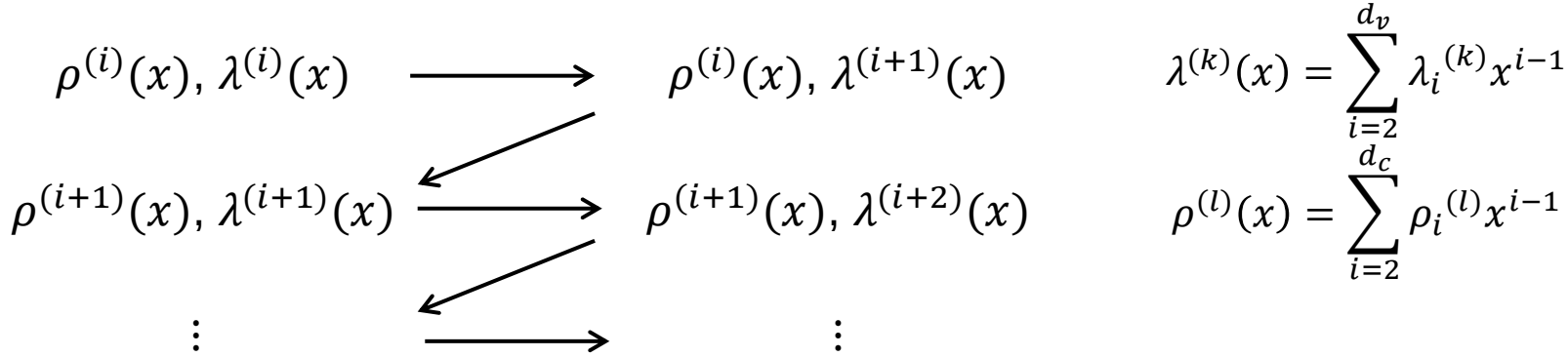
(II) Iterative Simplex Algorithm



(II) Iterative Simplex Algorithm : Convergence

We can see that R converge to R_{opt} with some iterations by simulation.

- We can proof the convergence of R since



If the LP solver guarantee that the solution is global optimum, then

$$R_{i,i+1} \geq R_{i,i} \geq R_{i-1,i} \quad \text{where} \quad R_{i,j} = 1 - \frac{\sum_l \frac{\rho_l{}^{(i)}}{l}}{\sum_k \frac{\lambda_k{}^{(j)}}{k}}$$

It means that the sequence of $R_{i,j}$ is non-decreasing.

Since $R_{i,j}$ is non-decreasing and $R \leq 1$, $R_{i,j}$ will converge to R_{opt}.

Conclusion

- **We describe differential evolution scheme to find good degree distribution.**
  - It gives good degree distribution candidates.
- **And we propose an iterative code rate optimizer.**
  - The iterative code rate optimizer (i.e. iterative simplex algorithm) increases code rate for given $\left(\frac{E_S}{N_0}\right)^*$.
- **Combined LDPC code design scheme.**
  - By simulation, we can see that proposed LDPC code design scheme has better $\left(\frac{E_b}{N_0}\right)^*$ and code rate than differential evolution only.

Conclusion : Simulation Results

	d_v	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8	ρ_6	ρ_7	$\left(\frac{E_b}{N_0}\right)_{dB}$	R
Richa-rdson	6	0.33241	0.24632	0.11014		0.31112			0.76611	0.23389	0.7997	0.50840
	8	0.30013	0.28395					0.41592	0.22919	0.77081	0.5778	0.50013
Diff. Evol. Only	6	0.31265	0.17269	0.10492	0.24427	0.16547			0.65951	0.34049	0.7836	0.49953
	8	0.30819	0.24250	0.00601	0.01471	0.00148	0.00856	0.41855	0.16636	0.83364	0.5628	0.49918
Propo-sed	6	0.34061	0.24400			0.41539			0.65951	0.34049	0.7288	0.50584
	8	0.30259	0.26274					0.43468	0.16636	0.83364	0.5619	0.49927