Unidimensional Scaling (UDS)
Multidimensional Scaling (MDS)

Álex Murillo*, Javier Trejos, Eduardo Piza,
Mario Villalobos y Alejandra Jimenez**

CIMPA - Universidad de Costa Rica
*Sede del Atlántico
**Instituto Tecnológico de Costa Rica

Índice

Planteamiento del Problema

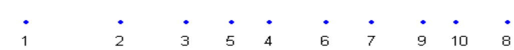
En este trabajo se abordan los problemas de

Unidimensional Scaling y
Multidimensional Scaling

métricos, para datos de disimilaridad simétricos, sin diferencias individuales, proponiendo algoritmos que mediante SA busca la optimización global.

UDS mediante SA

Dado un conjunto de estímulos y una matriz simétrica de disimilaridades entre ellos, UDS trata el problema de su representación mediante puntos en un continuo.



Ejemplo de datos *number.txt*

UDS mediante SA

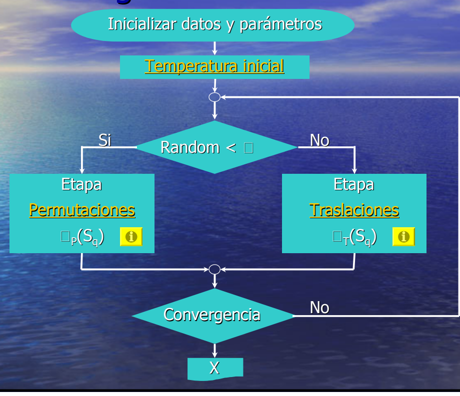
- El problema del UDS ha sido tratado en la literatura por diferentes autores y diferentes procedimientos de optimización han sido empleados para resolverlo.
- Entre los más utilizados ha sido el de Defays.

UDS mediante SA

Se presenta un algoritmo que, mediante SA, aborda el problema del UDS con una estrategia basada en un proceso aleatorio alternante ponderado que emplea *permutaciones* y *traslaciones* para encontrar la configuración óptima *X*, donde la función de pérdida a optimizar es:

$$S_q(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left| \delta_{ij} - \mid x_i - x_j \mid \right|^q$$

Algoritmo PerTSAUS



Comparación pertsaus2 vs sseuds2

	pertsaus2			sseuds2			
	Min $S_2(X)$	% Atr.	Tiempo	Min $S_2(X)$	□(Min)	% Atr.	Tiempo
numb	1.9599	100	0.17	1.9600	0.0001	60	1.04
d10	3.0064	85	0.18	3.0065	0.0001	95	1.06
d15	8.5294	85	0.32	8.5297	0.0003	90	1.73
d20	16.4181	100	0.47	16.4187	0.0006	95	2.42
d24	28.4915	40	0.63	28.4925	0.0010	55	3.03
d30	47.8620	30	0.88	47.8633	0.0013	10	3.99
morse	407.0136	35	1.25	407.0222	0.0086	50	5.37
d40	94.4261	20	1.41	94.4291	0.0030	20	5.73
d50	151.6201	5	2.01	151.6321	0.0120	5	7.63
d60	229.2092	30	2.75	229.2152	0.0060	10	9.71
d70	306.4005	5	3.62	306.4064	0.0059	5	12.06
d80	416.9796	75	4.48	416.9913	0.0117	10	14.42
d90	536.4617	5	5.62	536.4692	0.0075	5	16.96
d100	679.3477	5	6.75	679.3675	0.0198	5	19.71
d120	996.2261	10	9.48	996.2596	0.0335	5	25.72
d140	1369.7239	5	12.60	1369.7750	0.0511	5	32.95
d160	1825.9254	5	16.56	1825.9902	0.0648	5	43.09
d180	2339.4744	5	20.57	2339.5740	0.0996	5	53.08
d200	2870.0112	5	25.08	2870.1594	0.1482	5	63.38

Comparación pertsaus2 vs uniscalqa

	pertsaus2			uniscalqa			
	Min $S_2(X)$	% Atr.	Tiempo	Min $S_2(X)$	□(Min)	% Atr.	Tiempo
numb	1.9599	100	5.10	1.9599	0.0000	100	0.03
d10	3.0064	75	5.46	3.0064	0.0000	60	0.03
d15	8.5294	65	7.96	8.5294	0.0000	65	0.06
d20	16.4181	100	11.60	16.4181	0.0000	85	0.13
d24	28.4915	65	14.34	28.4915	0.0000	15	0.19
d30	47.8620	60	18.13	47.8620	0.0000	10	0.38
morse	407.0136	40	24.88	407.0136	0.0000	100	0.56
d40	94.4261	20	25.98	94.7192	0.2931	5	1.10
d50	151.6302	5	33.49	151.6922	0.0620	5	2.47
d60	229.2092	5	41.57	229.3148	0.1056	5	7.14
d70	306.4001	10	48.20	306.5159	0.1158	5	12.36
d80	416.9796	20	56.91	417.0640	0.0844	5	19.73
d90	536.5490	5	66.43	536.8108	0.2618	5	38.38
d100	679.3550	5	75.14	679.7925	0.4375	5	64.29
d120	996.2285	5	93.54	998.9198	2.6913	5	145.90
d140	1369.7253	5	116.01	1374.0494	4.3241	5	362.14
d160	1825.9387	5	138.39	1834.3846	8.4459	5	648.58
d180	2339.4920	5	159.81	2350.1170	10.6250	5	999.68
d200	2870.0170	5	184.32	2878.3301	8.3131	5	1518.76

Comparación pertsaus2 vs Defays

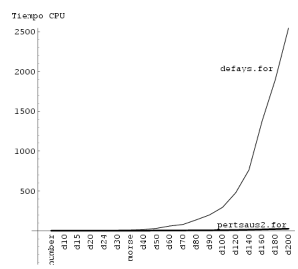


Figura 2.3: Gráfico del tiempo de CPU en segundos para los programas Defays.for y pertsaus2.for.

MDS mediante SA

Se propone un algoritmo que trata el problema de MDS en que ambas, la función de pérdida y la regla de composición pueden ser consideradas por cualquier métrica de **Minkowski**, usando un procedimiento aleatorio alternante compuesto por etapas de *permutaciones* y *traslaciones* en un contexto de SA.

$$S_q L_p E_r(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left| \delta_{ij}^r - d_{p,ij}^r \right|^q$$

MDS mediante SA

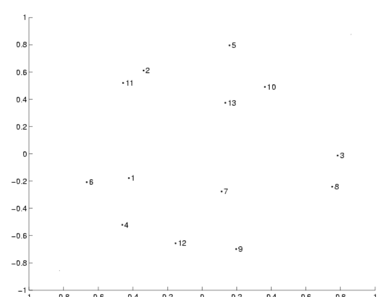
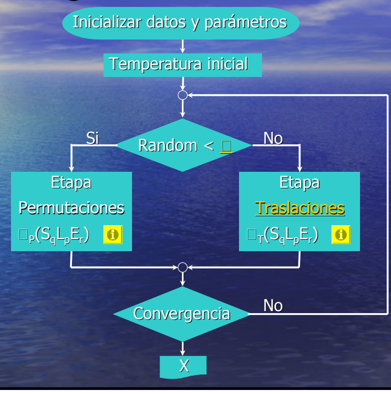


Figura 3.3: Representación de los datos de Funk, Horowitz, Lipshitz y Young (1974), en dimensión $\nu = 2$.

Algoritmo PTSAMDS



MDS mediante SA

Debido a que la eficacia de un algoritmo de SA depende en gran medida de los valores de los parámetros, ha sido llevado a cabo un estudio sobre varias funciones de pérdida para determinar los valores más convenientes de los principales parámetros.

Análisis de frecuencias en los principales parámetros en MDS

Parámetros				$S_1L_pE_1$							$S_2L_pE_1$						
α	τ	η	β	1	1.33	1.66	2	2.5	3	∞	1	1.33	1.66	2	2.5	3	∞
.05	1.05	.50	0.5	0	0	0	0	0	0	0	0	0	2	**2**	1	1	0
.05	1.05	.50	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.05	1.05	.95	0.5	2	0	0	27	0	0	7	24	10	23	**18**	11	7	6
.05	1.05	.95	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.05	1.50	.50	0.5	0	0	0	6	0	0	10	4	6	5	**11**	9	8	4
.05	1.50	.50	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.05	1.50	.95	0.5	98	65	60	52	60	70	31	57	39	35	**32**	30	34	51
.05	1.50	.95	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.50	1.05	.50	0.5	0	0	0	0	0	0	0	0	1	2	**2**	1	1	1
.50	1.05	.50	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.50	1.05	.95	0.5	1	35	25	2	10	30	11	9	20	20	**19**	23	36	22
.50	1.05	.95	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.50	1.50	.50	0.5	0	0	0	9	10	0	13	6	11	12	**12**	12	13	12
.50	1.50	.50	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0
.50	1.50	.95	0.5	0	0	15	4	20	0	29	1	13	1	**4**	13	1	4
.50	1.50	.95	1.0	0	0	0	0	0	0	0	0	0	0	**0**	0	0	0

Análisis comparativo de PTSAMDS

Tabla 3.6: Resultados en Fortran para *PTSAMDS.FOR* y *SPSS*, con $\nu = 2$.

	$S_2L_2E_1$				$S_2L_2E_2$			
	PTSAMDS.FOR			PROXSCAL	PTSAMDS.FOR			ALSCAL
Datos	**min**	**max**	**%**	resultados	**min**	**max**	**%**	resultados
colas	**0.0368**	0.0368	100	0.0418	**0.2341**	0.2397	45	0.2345
funk	**0.0603**	0.0627	90	0.0629	**0.3434**	0.3434	100	0.3457
d24	**0.1821**	0.1900	95	0.1823	**0.5709**	0.5709	100	0.5771
morse	**0.0901**	0.0901	100	0.0912	**0.4284**	0.4284	100	0.4337
d50	**0.2374**	0.2388	15	0.2431	**0.6593**	0.6600	65	0.6711
d70	**0.2581**	0.2582	75	0.2648	**0.6779**	0.6796	95	0.6900
d90	**0.2744**	0.2744	100	0.2820	**0.6974**	0.6982	95	0.7121
d120	**0.2925**	0.2936	15	0.2993	0.7132	0.7132	75	
d160	**0.3066**	0.3072	30	0.3136	0.7257	0.7265	45	
d200	**0.3146**	0.3149	60	0.3203	0.7356	0.7360	10	

Análisis comparativo de PTSAMDS con Smoothing, SMACOF y KYST para los datos de colas

p	PTSAMDS	Smoothing	SMACOF	KYST
1.00	**0.167546**	0.169437	0.218761	0.192295
1.33	**0.176408**	0.177194	0.178874	0.179519
1.66	**0.185259**	0.185316	0.186848	0.186276
2.00	**0.191779**	0.191782	0.191782	0.191978

MDS mediante SA

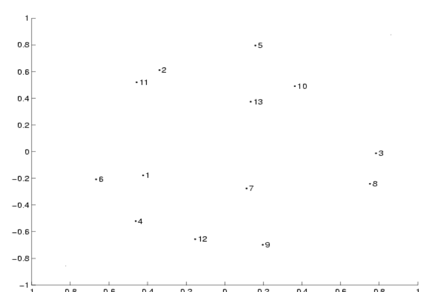


Figura 3.3: Representación de los datos de Funk, Horowitz, Lipshitz y Young (1974), en dimensión $\nu = 2$.

Conclusiones

- Se presenta el algoritmo PerTSAUS, para abordar el problema del UDS, el cual optimiza funciones de pérdida mínimo cuadráticas y desviaciones absolutas, mediante SA.
- Dicha estrategia se revela recomendable en la búsqueda de soluciones óptimas.
- Además, se observa que los métodos que siguen la estrategia propuesta por Defays no son recomendables para matrices de datos medianas o grandes.

Conclusiones

- Estas ideas se extienden al problema de MDS, proponiendo un modelo general para cualquier métrica de Minkowski, a la vez que se encuentran los valores óptimos de los principales parámetros del algoritmo PTSAMDS.
- Los resultados y las pruebas realizadas versus otros procedimientos de optimización, confirman la expectativa acerca de los beneficios de utilizar SA en los problemas de optimización de MDS.

Muchas gracias

Sobrecalentamiento Simulado

Condiciones para que la cadena de Markov sea ergódica aperiódica:

- conexidad entre los estados, i → j
- todos los vecindarios tienen el mismo cardinal,
- en un vecindario, todos los estados tienen la misma probabilidad de ser generados,
- reversibilidad entre los estados ($p_{ij}=p_{ji}$).

Procedimiento de Defays

Optimiza:

$$S_2(X) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\delta_{ij} - \mid x_i - x_j \mid\right)^2$$

Reparametrizando es equivalente a:

$$\sum_{i=1}^{n} (t_i^{(\hat\rho)})^2 = \max_{\rho \in \Psi(n)} \left\{ \sum_{i=1}^{n} (t_i^{(\rho)})^2 \right\},$$

$$\forall i = 1, \ldots, n,\ \hat{x}_i(\hat\rho) = t_i^{\hat\rho},$$

con,

$$t_i^{(\rho)} = (u_i^{(\rho)} - v_i^{(\rho)})/n, \quad \text{donde} \quad u_1^{(\rho)} = v_n^{(\rho)} = 0, \quad \text{y}$$

$$\forall\, i \geq 2,\ u_i^{(\rho)} = \sum_{j=1}^{i-1} \delta_{\rho(i)\rho(j)}; \quad \forall\, i < n,\ v_i^{(\rho)} = \sum_{j=i+1}^{n} \delta_{\rho(i)\rho(j)}.$$

Algoritmo Temperatura Inicial de PerTSAUS

Inicializar: Ma, $Ma_{\text{máx}}$, χ
$Promediar \leftarrow 0$
$cont \leftarrow 0$
$Prueba \leftarrow 1$
Repetir
  Generar $\hat{X}^{(s+1)} \in \vartheta(X, A, V)$, aleatoriamente
  $dt \leftarrow \triangle_T(S_q)$
  Si $dt < 0$ **Entonces**
    $Promediar \leftarrow (Promediar + dt)$
    $cont \leftarrow (cont + 1)$
  Fin(Si)
  $Prueba \leftarrow (Prueba + 1)$
Hasta $(Prueba = Ma_{\text{máx}})$ o $(cont = Ma)$
$T_0 \leftarrow \dfrac{Promediar/cont}{\ln(\chi)}$

CIMPA-UCR

Etapa de Permutaciones Algoritmo de PerTSAUS

Para 1 **hasta** LC **Hacer**
  Generar una transposicion $\hat{X}^{(s+1)}$ de X
  $dp \leftarrow \triangle_P(S_q)$
  Si $dp > 0$ **Entonces**
    $X \leftarrow \hat{X}^{(s+1)}$
    $S \leftarrow (S - dp)$
  Fin(Si)
Fin(Para)

CIMPA-UCR

Etapa Traslaciones PerTSAUS

Para 1 **hasta** LC **Hacer**
  Generar $\hat{X}^{(s+1)} \in \vartheta(X, A, V)$, trasladando x_l
  $dt \leftarrow \triangle_T(S_q)$
  Si $dt > 0$ **Entonces**
    $X \leftarrow \hat{X}^{(s+1)}$
    $S \leftarrow (S - dt)$
    $V(l) \leftarrow \Big(V(l) + \eta * (\text{máx}\,\Delta - V(l))\Big)$
  Sino
    Actualizar A, tal que $A(l) \leftarrow -A(l)$
    $V(l) \leftarrow \Big(V(l) - \eta * V(l)\Big)$
    Si $(\exp(dt/T) > Aleatorio(0,1))$ **Entonces**
      $X \leftarrow \hat{X}^{(s+1)}$
      $S \leftarrow (S - dt)$
    Fin(Si)
  Fin(Si)
Fin(Para)
$T \leftarrow (T * \gamma)$

CIMPA-UCR

Gráfico función de pérdida

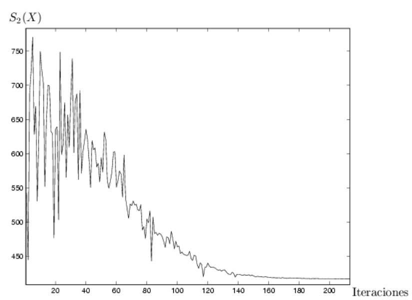
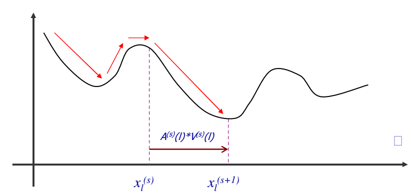


Figura 2.1: Gráfico de la función de pérdida (2.1) en cada iteración de una ejecución del programa pertsaus2.for para los datos d80.txt.

CIMPA-UCR

Vecindario en PerTSAUS

$$\hat{x}_l^{(s+1)} = \hat{x}_l^{(s)} + A^{(s)}(l) * V^{(s)}(l)$$



CIMPA-UCR

Fórmulas de diferencia lineales en etapa de permutaciones

$$\begin{aligned}\triangle(S_q) &= S_q(\hat{X}^{(s)}) - S_q(\hat{X}^{(s+1)}) \\ &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\left(\varphi_{ij,j}^{(s)}\right)^q - \left(\varphi_{ij,j}^{(s+1)}\right)^q \right),\end{aligned}$$

siendo, $\forall l, m, k = 1, \ldots, n$ y $q = 1, 2$,

$$\left(\varphi_{lm,k}^{(s)}\right)^q = \left|\delta_{lm} - \mid \hat{x}_l^{(s)} - \hat{x}_k^{(s)} \mid\right|^q = \left|\delta_{lm} - d_{l^{(s)}k^{(s)}}\right|^q.$$

$$\triangle_P(S_1) = \sum_{\substack{i=1 \\ i\neq k, i\neq l}}^{n} \left(\varphi_{ik,k}^{(s)} + \varphi_{il,l}^{(s)} - \varphi_{ik,l}^{(s)} - \varphi_{il,k}^{(s)}\right)$$

$$\begin{aligned}\triangle_P(S_2) &= 2 \sum_{\substack{i=1 \\ i\neq k, i\neq l}}^{n} (\delta_{ik} - \delta_{il}) \left(d_{i^{(s)}l^{(s)}} - d_{i^{(s)}k^{(s)}}\right) \\ &= 2\,(\delta_k - \delta_l) \left(d_{(l)} - d_{(k)}\right) + 4\delta_{lk} d_{lk}.\end{aligned}$$

CIMPA-UCR

Fórmulas de diferencia lineales en etapa de traslaciones

$$\triangle_T(S_q) = \sum_{\substack{i=1 \\ i\neq l}}^{n} \left(\mid \delta_{il} - d_{i^{(s)}l^{(s)}} \mid^q - \mid \delta_{il} - d_{i^{(s)}l^{(s+1)}} \mid^q\right).$$

Además, si $q = 2$ se tiene para los lenguajes orientados al cálculo matricial

$$\triangle_T(S_2) = d^2_{l^{(s)}l^{(s+1)}} + 2\delta_l (d_{l^{(s+1)}} - d_{(l)})^t + d_l d_l^t - d_{l^{(s+1)}} d^t_{l^{(s+1)}},$$

donde $d_{l^{(s+1)}}$ denota el vector:

$$d_{l^{(s+1)}} = \left(d_{l^{(s+1)}1^{(s)}}, \ldots, d_{l^{(s+1)}n^{(s)}}\right)^t.$$

Comparación pertsaus2 vs tsaus2

	pertsaus2.for				tsaus2.for			
	min	max	atr	tiempo	min	max	atr	tiempo
Datos	(S_2)	(S_2)	%	CPU	(S_2)	(S_2)	%	CPU
numb	1.9599	**1.9599**	**100**	**0.17**	1.9599	1.9623	60	0.39
d10	3.0064	3.1633	**85**	**0.18**	3.0064	**3.1433**	60	0.40
d15	8.5294	**8.5302**	**85**	**0.32**	8.5294	8.5343	55	0.65
d20	16.4181	**16.4181**	**100**	**0.47**	16.4181	16.4238	90	0.94
d24	28.4915	29.0921	40	**0.63**	28.4915	**28.7779**	**45**	1.20
d30	47.8620	48.8536	**30**	**0.88**	47.8620	**47.8862**	5	1.62
morse	407.0136	**407.2327**	**35**	**1.25**	407.0136	412.9899	30	2.19
d40	94.4261	94.8749	20	**1.41**	94.4261	**94.5113**	**40**	2.44
d50	**151.6201**	**151.9794**	**5**	**2.01**	151.6278	152.0772	0	3.39
d60	229.2092	230.2471	**30**	**2.75**	229.2092	**230.2355**	15	4.47
d70	306.4005	**306.4446**	5	**3.62**	306.4005	306.5401	5	5.72
d80	416.9796	**416.9982**	**75**	**4.48**	416.9796	422.0376	5	7.05
d90	**536.4617**	**536.9662**	**5**	**5.62**	536.4739	537.1609	0	8.56
d100	**679.3477**	**682.2167**	**5**	**6.75**	679.3604	683.2681	0	10.24
d120	**996.2261**	998.8536	**10**	**9.48**	996.2278	**998.2338**	0	13.92
d140	**1369.7239**	1374.1523	**5**	**12.60**	1369.7304	**1372.3284**	0	18.20
d160	**1825.9254**	1831.2898	**5**	**16.56**	1825.9430	**1831.2857**	0	23.07
d180	**2339.4744**	2345.7609	**5**	**20.57**	2339.5847	**2345.2336**	0	28.40
d200	**2870.0112**	**2871.0376**	**5**	**25.08**	2870.0699	2873.1551	0	34.41

Fórmulas diferencia lineales en etapa permutaciones PTSAMDS

$$\begin{aligned}\triangle_P(S_qL_pE_r) &= S_qL_pE_r(\hat{X}^{(s)}) - S_qL_pE_r(\hat{X}^{(s+1)})\\ &= \sum_{\substack{i=1\\ i\neq k, i\neq l}}^{n} \left(\left(\varphi_{ik,k}^{(s)pr}\right)^q + \left(\varphi_{il,l}^{(s)pr}\right)^q - \left(\varphi_{ik,l}^{(s)pr}\right)^q - \left(\varphi_{il,k}^{(s)pr}\right)^q\right),\end{aligned} \tag{3.2}$$

donde,

$$\left(\varphi_{lm,k}^{(s)pr}\right)^q = \left|\delta_{lm}^r - d_{p,l^{(s)}k^{(s)}}^r\right|^q, \qquad \forall l, m, k = 1, \ldots, n.$$

Resulta eficiente para la implementación en lenguajes orientados al cálculo matricial como MatLab o Mathematica, disponer de una expresión matricial eficiente de $\triangle(S_2L_pE_r)$. Así, para la etapa de permutaciones, se tiene

$$\triangle_P(S_2L_pE_r) = 2\left(\delta_k^r - \delta_l^r\right)\left(d_{p,(l)}^r - d_{p,(k)}^r\right) + 4\delta_{lk}^r d_{p,lk}^r, \tag{3.3}$$

Fórmulas diferencia lineales en etapa traslaciones PTSAMDS

$$\triangle_T(S_qL_pE_r) = \sum_{\substack{i=1\\ i\neq l}}^{n}\left(\left|\delta_{il}^r - d_{p,i^{(s)}l^{(s)}}^r\right|^q - \left|\delta_{il}^r - d_{p,i^{(s)}l^{(s+1)}}^r\right|^q\right). \tag{3.4}$$

Además, se tiene para los lenguajes orientados al cálculo matricial,

$$\begin{aligned}\triangle_T(S_2L_pE_r) &= d_{p,l^{(s)}l^{(s+1)}}^{2r} + 2\delta_l^r(d_{p,l^{(s+1)}}^r - d_{p,(l)}^r)^t\\ &\quad + d_{p,l}^r(d_{p,l}^r)^t - d_{p,l^{(s+1)}}^r(d_{p,l^{(s+1)}}^r)^t,\end{aligned} \tag{3.5}$$

donde $d_{p,l^{(s+1)}}^r$ denota el vector:

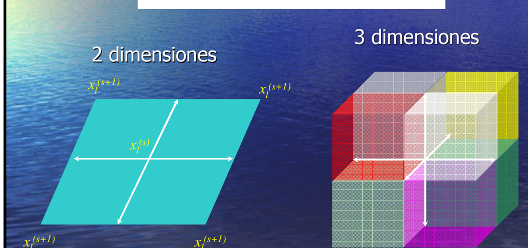
$$d_{p,l^{(s+1)}}^r = \left(d_{p,l^{(s+1)}1^{(s)}}^r, \ldots, d_{p,l^{(s+1)}n^{(s)}}^r\right)^t.$$

Etapa Traslaciones PTSAMDS

Para 1 **hasta** LC **Hacer**
  Generar $\hat{X}^{(s+1)} \in \vartheta(X, V)$, al trasladar x_l
  $dt \leftarrow \triangle_T(S_qL_pE_r)$
  Si $dt > 0$ **Entonces**
    $X \leftarrow \hat{X}^{(s+1)}$
    $S \leftarrow (S - dt)$
    $V(l,j) \leftarrow (V(l,j) * \tau)$, para $j = 1, \ldots, \nu$
  Sino
    Para j=1 **hasta** ν **Hacer**
      Si $Aleatorio(0,1) < \beta$ **Entonces** $V(l,j) \leftarrow -V(l,j)$
      $V(l,j) \leftarrow (V(l,j) * \eta)$
    Fin(Para)
    Si $\exp(dt/T) > Aleatorio(0,1)$ **Entonces**
      $X \leftarrow \hat{X}^{(s+1)}$
      $S \leftarrow (S - dt)$
    Fin(Si)
  Fin(Si)
Fin(Para)
$T \leftarrow (T * \gamma)$

Vecindario en PTSAMDS

$$\hat{x}_{lj}^{(s+1)} = \hat{x}_{lj}^{(s)} + V^{(s)}(l,j).$$



Cálculo de frecuencias del óptimo para $S_2L_2E_1$ en dimensión 2

Parámetros													
α	τ	η	β	**colas**	**funk**	**d24**	**morse**	**d50**	**d70**	**d90**	**d120**	**d160**	**d200**
.05	1.05	.50	0.5	7	3	3	3	0	0	0	0	0	0
.05	1.05	.50	1.0	0	0	0	0	0	0	0	0	0	0
.05	1.05	.95	0.5	18	10	20	18	6	13	18	2	1	0
.05	1.05	.95	1.0	0	0	0	0	0	0	0	0	0	0
.05	1.50	.50	0.5	15	6	12	10	1	10	16	0	2	0
.05	1.50	.50	1.0	0	0	0	0	0	0	0	0	0	0
.05	**1.50**	**.95**	**0.5**	**16**	**11**	**20**	**20**	**5**	**16**	**20**	**2**	**5**	**1**
.05	1.50	.95	1.0	0	0	0	0	0	0	0	0	0	0
.50	1.05	.50	0.5	8	3	5	0	0	0	0	0	0	0
.50	1.05	.50	1.0	0	0	0	0	0	0	0	0	0	0
.50	1.05	.95	0.5	18	11	13	16	7	13	18	2	2	0
.50	1.05	.95	1.0	0	0	0	0	0	0	0	0	0	0
.50	1.50	.50	0.5	17	5	15	14	4	13	16	0	1	0
.50	1.50	.50	1.0	0	0	0	0	0	0	0	0	0	0
.50	1.50	.95	0.5	5	12	0	1	0	0	0	1	0	0
.50	1.50	.95	1.0	0	0	0	0	0	0	0	0	0	0
Suma				104	61	88	82	23	65	88	7	11	1

Parámetros														
α	τ	η	β	**colas**	**funk**	**d24**	**morse**	**d50**	**d70**	**d90**	**d120**	**d160**	**d200**	**Promedio**
.05	**1.50**	**.95**	**0.5**	0.15	0.18	0.23	0.24	0.22	0.25	0.23	0.29	0.45	1.00	**0.32**