

# Algoritmo Poblacional Cooperativo

Presentación de Metaheurística Original



# Algoritmo Poblacional Cooperativo

El **APC** es una metaheurística inspirada en una sociedad **comunista**, en vez de simular un comportamiento competitivo como muchos algoritmos de poblaciones se pretende **simular cooperación** entre la población.

Partiendo del aforismo, 'De cada cual según sus capacidades, a cada cual según sus necesidades', los individuos **aportarán sus capacidades** (partes que más contribuyan a la función objetivo) y **cubrirán sus necesidades** sustituyendo las partes con menor contribución por las aportaciones de otras soluciones.

Intentando potenciar este comportamiento se ha ido mejorando el comportamiento del algoritmo con diferentes componentes.

# Componentes del Algoritmo

# Inicialización de una Población

- **Simulando una población de personas** crearemos un conjunto de soluciones.
- Cada solución será **inicializada de manera aleatoria** obteniendo siempre una solución válida.
- Guardamos también una estimación de **cuánto contribuye** cada elemento o **parte de una solución** a la **calidad de esta**.

# Explotación: Copia de Mejores a Peores

- Los elementos que más contribuyen a su solución se toman como **aportaciones a la sociedad** y los peores elementos, **como las necesidades a satisfacer**.
- Las necesidades intentan ser sustituidas por aportaciones **si estas mejoran a la solución**.
- Hay una **alta probabilidad** de mejorar las soluciones, por lo que en **pocas evaluaciones** de la función objetivo **converge más rápidamente** (aunque perdiendo algo de diversidad)

# Explotación: Búsqueda Local Limitada

- Tras algo de explotación eficiente por la copia aplicamos una búsqueda local para acercarnos más al óptimo local y **considerar soluciones vecinas** que **no se puedan formar** con elementos de la población.
- Se establece como **criterio de parada** un número de **evaluaciones seguidas sin mejora**.
- Combinado con la primera componente se llega al criterio de parada antes y se **aprovechan mejor** las evaluaciones disponibles.

# Exploración: Re-Inicialización o Mutación

- Para mantener un **equilibrio** explotación / exploración es necesario **aportar una diversidad fuerte**, ya que la copia disminuye esta.
- Se modifican las soluciones **empezando por las mejores**, para evitar quedarnos con soluciones en óptimos locales y que reduzcan la diversidad de la población al **esparcirse sus elementos** con la copia de mejores.
- Se prueban dos opciones, **re-inicializar las soluciones** completamente o **mutarlas fuertemente**.

# Estructura del Algoritmo

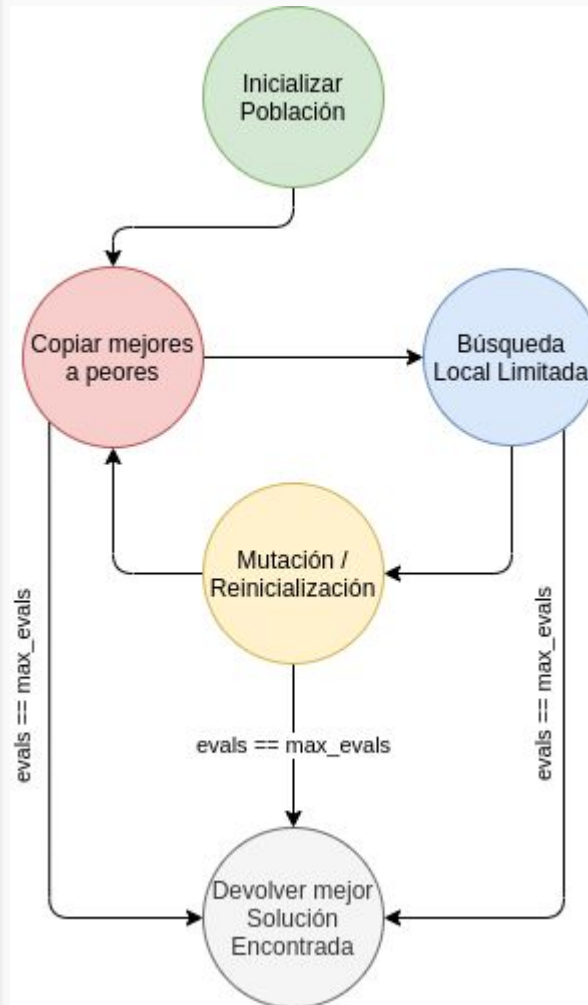
Combinación de las Componentes



# Estructura APC

El APC posee una **estructura simple**, lo que permitirá **analizar fácilmente** su funcionamiento y **adaptarlo a cada problema**, posiblemente con un **tiempo de ejecución reducido**.

Además, presenta oportunidades para una **ejecución en paralelo**, ya sea en la ejecución de **las búsquedas locales** de cada solución ó con varias **poblaciones independientes** que cada cierto periodo compartan información.



```
inicializarPoblacion()
```

```
evals = filas de fitness
```

```
Mientras evals < max_evals
```

```
    evals += copyBestWorst()
```

```
    Para un porcentaje de sol en las mejores soluciones:
```

```
        evals += lsbf(sol, max_evals=max_evals - evals)
```

```
    Para un porcentaje de sol en las mejores soluciones:
```

```
        sol = mutarSolucion(sol) ó inicializarSolucion()
```

```
        Recalculamos fitness para sol y lo contamos en evals
```

```
Devolver mejor solución encontrada hasta el momento
```

# Análisis Experimental

Resultados de Aplicar APC en el MDP

**Algoritmo Poblacional Cooperativo – Mutación**

Caso	Coste Obtenido	Desv	Tiempo
GKD-c_11_n500_m50	19587,12018	0,00	0,55992436
GKD-c_12_n500_m50	19360,22393	0,00	0,5472858
GKD-c_13_n500_m50	19366,69852	0,00	0,57022882
GKD-c_14_n500_m50	19458,56466	0,00	0,56702209
GKD-c_15_n500_m50	19422,1464	0,00	0,56954384
GKD-c_16_n500_m50	19680,20477	0,00	0,56529546
GKD-c_17_n500_m50	19331,38841	0,00	0,5716064
GKD-c_18_n500_m50	19461,3946	0,00	0,57317019
GKD-c_19_n500_m50	19477,32577	0,00	0,55873322
GKD-c_20_n500_m50	19604,84356	0,00	0,5608952
MDG-b_1_n500_m50	777890,27	0,02	0,55518985
MDG-b_2_n500_m50	779963,54	0,00	0,5518415
MDG-b_3_n500_m50	774256,41	0,32	0,56641173
MDG-b_4_n500_m50	774284,6	0,14	0,53820705
MDG-b_5_n500_m50	773236,45	0,31	0,53969979
MDG-b_6_n500_m50	774909,14	0,03	0,54512715
MDG-b_7_n500_m50	775564,88	0,21	0,53473687
MDG-b_8_n500_m50	779168,62	0,00	0,55023479
MDG-b_9_n500_m50	772348,09	0,32	0,5414598
MDG-b_10_n500_m50	773966,97	0,13	0,54110074
MDG-a_31_n2000_m200	113440	0,61	0,78299785
MDG-a_32_n2000_m200	113512	0,51	0,71738744
MDG-a_33_n2000_m200	113641	0,42	0,75717759
MDG-a_34_n2000_m200	113338	0,76	0,75634098
MDG-a_35_n2000_m200	113317	0,76	0,75610328
MDG-a_36_n2000_m200	113689	0,49	0,78806853
MDG-a_37_n2000_m200	113330	0,77	0,74461126
MDG-a_38_n2000_m200	113782	0,52	0,73512053
MDG-a_39_n2000_m200	113309	0,78	0,75198364
MDG-a_40_n2000_m200	113605	0,51	0,73424077
			0,254021025 0,621058218

**Algoritmo Poblacional Cooperativo – Reinicialización**

Caso	Coste Obtenido	Desv	Tiempo
GKD-c_11_n500_m50	19587,12018	0,00	0,56175947189331
GKD-c_12_n500_m50	19360,22393	0,00	0,565002918243408
GKD-c_13_n500_m50	19366,69852	0,00	0,571977615356445
GKD-c_14_n500_m50	19458,56466	0,00	0,562175989151001
GKD-c_15_n500_m50	19422,1464	0,00	0,555885791778564
GKD-c_16_n500_m50	19680,20477	0,00	0,567338705062866
GKD-c_17_n500_m50	19331,38841	0,00	0,583657503128052
GKD-c_18_n500_m50	19461,3946	0,00	0,567113637924194
GKD-c_19_n500_m50	19477,32577	0,00	0,560760021209717
GKD-c_20_n500_m50	19604,84356	0,00	0,583845376968384
MDG-b_1_n500_m50	772255,95	0,74	0,564834594726563
MDG-b_2_n500_m50	776550,08	0,44	0,55630898475647
MDG-b_3_n500_m50	771284,78	0,71	0,543460369110107
MDG-b_4_n500_m50	770039,81	0,69	0,545430183410645
MDG-b_5_n500_m50	774093,45	0,20	0,54151725769043
MDG-b_6_n500_m50	772260,48	0,37	0,539308547973633
MDG-b_7_n500_m50	774913,07	0,30	0,541873693466187
MDG-b_8_n500_m50	773724,51	0,70	0,551995038986206
MDG-b_9_n500_m50	770891,29	0,50	0,548358678817749
MDG-b_10_n500_m50	770718,83	0,55	0,536827325820923
MDG-a_31_n2000_m200	113044	0,96	0,786602735519409
MDG-a_32_n2000_m200	113192	0,79	0,821762084960938
MDG-a_33_n2000_m200	113029	0,96	0,799296855926514
MDG-a_34_n2000_m200	113101	0,96	0,776726007461548
MDG-a_35_n2000_m200	113018	1,02	0,759539365768433
MDG-a_36_n2000_m200	113395	0,75	0,818631649017334
MDG-a_37_n2000_m200	113101	0,97	0,758532047271729
MDG-a_38_n2000_m200	113257	0,98	0,733278751373291
MDG-a_39_n2000_m200	112800	1,23	0,740627288818359
MDG-a_40_n2000_m200	113049	1,00	0,780584573745728
			0,49386313 0,630833768844605

### Algoritmo Poblacional Cooperativo – Mutación(50.000 evals)

Caso	Coste Obtenido	Desv	Tiempo
GKD-c_11_n500_m50	19587,12018	0,00	0,289173603
GKD-c_12_n500_m50	19360,22393	0,00	0,284993887
GKD-c_13_n500_m50	19366,69852	0,00	0,290518999
GKD-c_14_n500_m50	19458,56466	0,00	0,292820692
GKD-c_15_n500_m50	19422,1464	0,00	0,287569284
GKD-c_16_n500_m50	19680,20477	0,00	0,298054218
GKD-c_17_n500_m50	19331,38841	0,00	0,29076314
GKD-c_18_n500_m50	19461,3946	0,00	0,28792429
GKD-c_19_n500_m50	19477,32577	0,00	0,279278278
GKD-c_20_n500_m50	19604,84356	0,00	0,285269976
MDG-b_1_n500_m50	771292,83	0,87	0,273779631
MDG-b_2_n500_m50	779963,54	0,00	0,309068441
MDG-b_3_n500_m50	769359,28	0,95	0,279470921
MDG-b_4_n500_m50	774284,6	0,14	0,283094168
MDG-b_5_n500_m50	773236,45	0,31	0,276847839
MDG-b_6_n500_m50	772896,06	0,29	0,282467127
MDG-b_7_n500_m50	766289,61	1,41	0,288223743
MDG-b_8_n500_m50	775496,75	0,47	0,279997349
MDG-b_9_n500_m50	769699,54	0,66	0,274867773
MDG-b_10_n500_m50	772946,9	0,26	0,280899048
MDG-a_31_n2000_m200	113244	0,78	0,3983953
MDG-a_32_n2000_m200	113220	0,76	0,381247997
MDG-a_33_n2000_m200	113452	0,59	0,395916462
MDG-a_34_n2000_m200	112889	1,15	0,394384861
MDG-a_35_n2000_m200	113045	0,99	0,414612055
MDG-a_36_n2000_m200	113589	0,58	0,406620264
MDG-a_37_n2000_m200	113054	1,01	0,410608053
MDG-a_38_n2000_m200	113608	0,67	0,399129868
MDG-a_39_n2000_m200	112913	1,13	0,406035185
MDG-a_40_n2000_m200	113545	0,57	0,406095743

0,453402927 0,32427094

### Algoritmo Poblacional Cooperativo – Mutación(300.000 evals)

Caso	Coste Obtenido	Desv	Tiempo
GKD-c_11_n500_m50	19587,12018	0,00	1,702629566
GKD-c_12_n500_m50	19360,22393	0,00	1,652034283
GKD-c_13_n500_m50	19366,69852	0,00	1,712421656
GKD-c_14_n500_m50	19458,56466	0,00	1,754395485
GKD-c_15_n500_m50	19422,1464	0,00	1,719398975
GKD-c_16_n500_m50	19680,20477	0,00	1,778457403
GKD-c_17_n500_m50	19331,38841	0,00	1,68274188
GKD-c_18_n500_m50	19461,3946	0,00	1,733003139
GKD-c_19_n500_m50	19477,32577	0,00	1,705844402
GKD-c_20_n500_m50	19604,84356	0,00	1,701651096
MDG-b_1_n500_m50	778030,57	0,00	1,705351114
MDG-b_2_n500_m50	779963,54	0,00	1,712192059
MDG-b_3_n500_m50	775511,98	0,16	1,741528511
MDG-b_4_n500_m50	774957,58	0,06	1,649224281
MDG-b_5_n500_m50	775610,96	0,00	1,66425705
MDG-b_6_n500_m50	775081,04	0,01	1,659719229
MDG-b_7_n500_m50	777232,88	0,00	1,653734684
MDG-b_8_n500_m50	779168,62	0,00	1,666186571
MDG-b_9_n500_m50	772348,09	0,32	1,643175364
MDG-b_10_n500_m50	774259,92	0,09	1,672258139
MDG-a_31_n2000_m200	113745	0,35	2,283876896
MDG-a_32_n2000_m200	113715	0,33	2,359479189
MDG-a_33_n2000_m200	113914	0,18	2,36187911
MDG-a_34_n2000_m200	113606	0,52	2,339020729
MDG-a_35_n2000_m200	113596	0,51	2,271746397
MDG-a_36_n2000_m200	114181	0,06	2,278388023
MDG-a_37_n2000_m200	113494	0,63	2,311555624
MDG-a_38_n2000_m200	113891	0,43	2,28896904
MDG-a_39_n2000_m200	113466	0,64	2,287474394
MDG-a_40_n2000_m200	113836	0,31	2,282280445

0,153358944 1,899162491

### Algoritmo Poblacional Cooperativo – Mutación Sin Copia

Caso	Coste Obtenic	Desv	Tiempo
GKD-c_11_n500_m50	19587,12018	0,00	0,533421755
GKD-c_12_n500_m50	19360,22393	0,00	0,533498287
GKD-c_13_n500_m50	19366,69852	0,00	0,541678429
GKD-c_14_n500_m50	19458,56466	0,00	0,539567709
GKD-c_15_n500_m50	19422,1464	0,00	0,522727013
GKD-c_16_n500_m50	19680,20477	0,00	0,529358625
GKD-c_17_n500_m50	19331,38841	0,00	0,533056736
GKD-c_18_n500_m50	19461,3946	0,00	0,534056902
GKD-c_19_n500_m50	19477,32577	0,00	0,534234762
GKD-c_20_n500_m50	19604,84356	0,00	0,53540206
MDG-b_1_n500_m50	770678,38	0,94	0,542438269
MDG-b_2_n500_m50	769894,29	1,29	0,543371677
MDG-b_3_n500_m50	774747,09	0,26	0,537825823
MDG-b_4_n500_m50	774821,09	0,07	0,527902126
MDG-b_5_n500_m50	774953,21	0,08	0,522265434
MDG-b_6_n500_m50	774864,58	0,04	0,530363083
MDG-b_7_n500_m50	776405,36	0,11	0,530771494
MDG-b_8_n500_m50	771154,04	1,03	0,530710936
MDG-b_9_n500_m50	773102,53	0,22	0,52880621
MDG-b_10_n500_m50	772573,27	0,31	0,535687923
MDG-a_31_n2000_m200	113504	0,56	0,734916687
MDG-a_32_n2000_m200	113626	0,41	0,757445574
MDG-a_33_n2000_m200	113461	0,58	0,763593674
MDG-a_34_n2000_m200	113389	0,71	0,755260944
MDG-a_35_n2000_m200	113424	0,66	0,757683754
MDG-a_36_n2000_m200	113924	0,29	0,769475937
MDG-a_37_n2000_m200	113370	0,74	0,836158514
MDG-a_38_n2000_m200	113797	0,51	0,742139339
MDG-a_39_n2000_m200	113716	0,42	0,784792662
MDG-a_40_n2000_m200	113566	0,55	0,715050459

0,32602839368847 0,609455427

Algoritmo	Desv	Tiempo
<u>Greedy</u>	1,365399	0,003558
<u>BL</u>	0,947056	0,2784
<u>Greedy + BL</u>	0,990486	0,258979
<u>AGG UO</u>	1,118993	30,22042
<u>AGG UE</u>	0,809609	38,34096
<u>AGG P</u>	2,746465	13,30146
<u>AGE UO</u>	1,102157	21,03321
<u>AGE UE</u>	1,017798	25,99481
<u>AGE P</u>	1,460289	20,40478
<u>AM_10_1_UE</u>	0,926886	1,740556
<u>AM_10_01_UE</u>	0,827351	6,189337
<u>AM_10_01mej_UE</u>	0,789479	2,003546
<u>AM_10_1_UO</u>	1,091818	1,971661
<u>AM_10_01_UO</u>	1,11749	1,98165
<u>AM_10_01mej_UO</u>	0,745433	7,436128
<u>ES</u>	1,321724	0,56407
<u>BMB</u>	0,726644	0,580085
<u>ILS</u>	0,528243	0,518678
<u>ILS-ES</u>	0,856457	1,125062
<u>APC-M</u>	0,254021	0,621058
<u>APC-R</u>	0,493863	0,630834

Izquierda: Mejor versión sin la componente de copia.

Derecha: Comparación con otras metaheurísticas



Espero que os  
haya resultado  
interesante :)

Fco Javier Bolívar Expósito  
2019 - 2020

