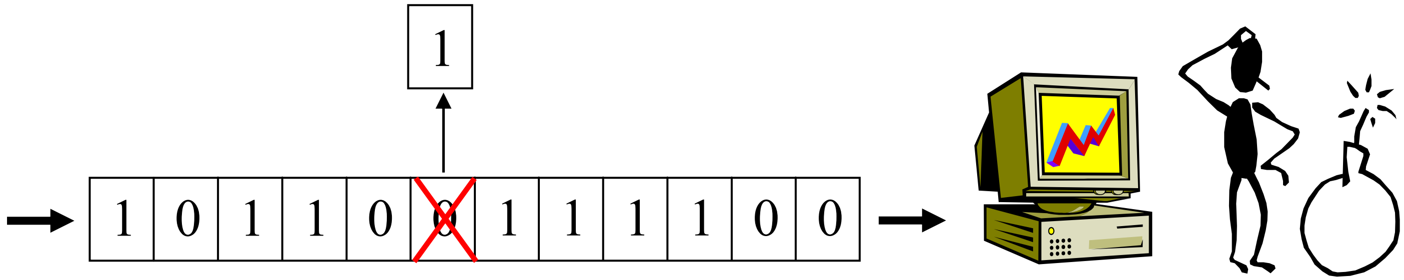
Solving the Error Correcting Code Problem with Parallel Hybrid Heuristics

Lenguajes y Ciencias de la Computación

Enrique Alba and J. Francisco Chicano

Introduction

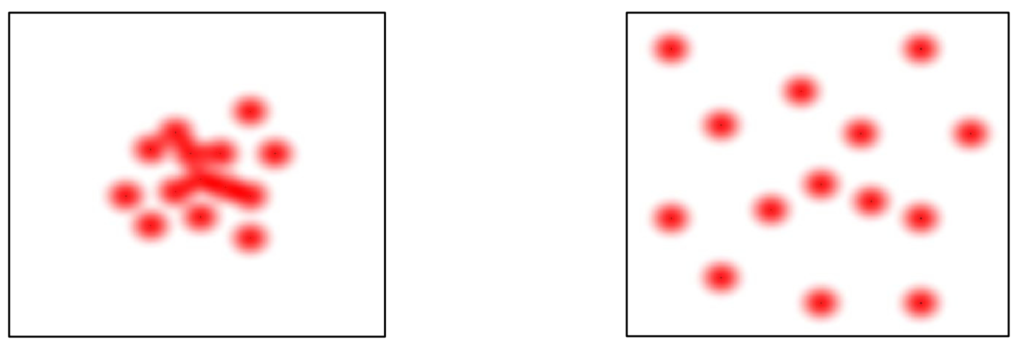
- Some applications cannot afford the resubmission of an erroneous msg



- Linear Block Codes

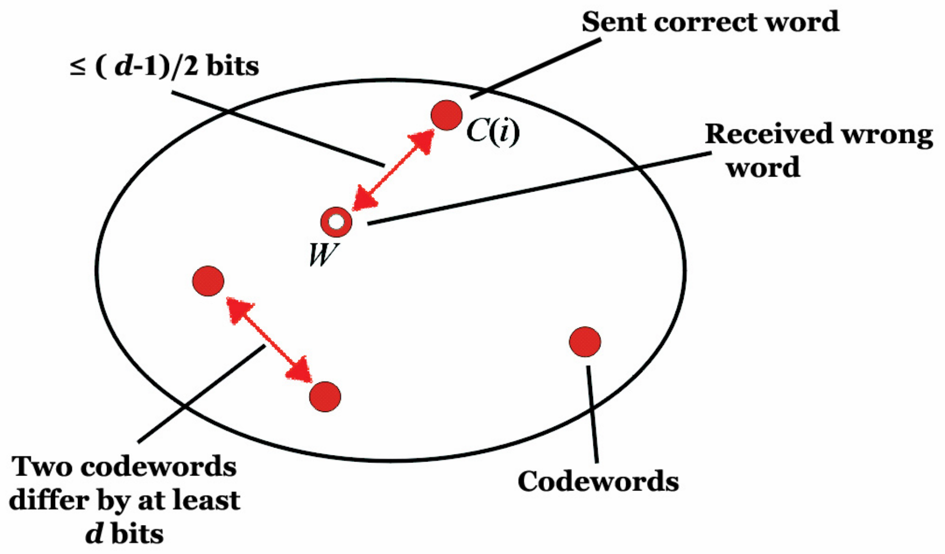
1	0	1	1	0	0	1	1	1	1	0	0	1	0	1	1	0	0

- Larger (Hamming) distance → Larger autocorrection capacity



ECC Problem

- Designing error correcting codes (*n*, *M*, *d*) is NP-hard
- Objective: Given *n* and *M* find a code that maximizes *d*



- Instance: n = 12 bits, M = 24 words
- Maximum *d*: 6

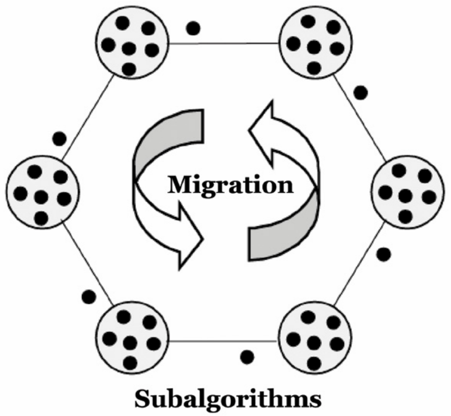
Genetic Algorithms

Introduction
ECC Problem
Algorithms
GA
RA
Hybrid
Experiments
Conclusions & Future Work

- **Pseudo-code of a general GA**

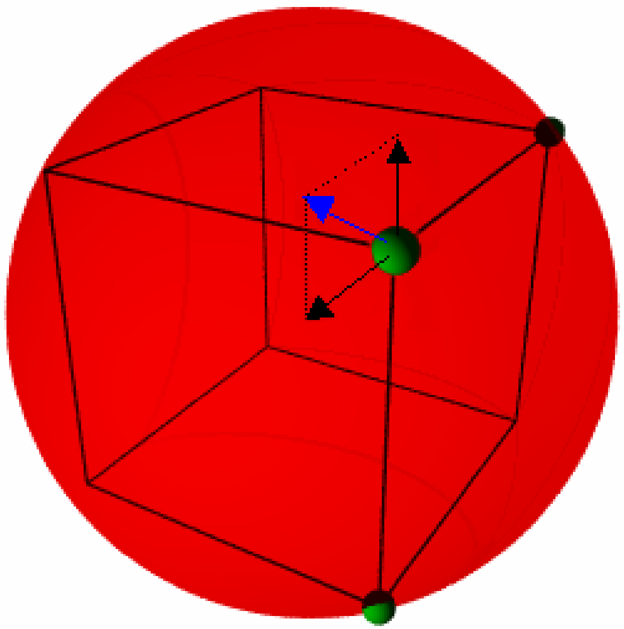
$t := 0;$
initialize: $P(0) := \{\vec{a}_1(0), \ldots, \vec{a}_\mu(0)\} \in I^\mu;$
evaluate: $P(0) : \{\Phi(\vec{a}_1(0)), \ldots, \Phi(\vec{a}_\mu(0))\};$
while $\iota(P(t)) \neq$ **true do** //Reproductive loop
  select: $P'(t) := s_{\Theta_s}(P(t));$
  recombine: $P''(t) := \otimes_{\Theta_c}(P'(t));$
  mutate: $P'''(t) := m_{\Theta_m}(P''(t));$
  evaluate: $P'''(t) : \{\Phi(\vec{a}_1'''(t)), \ldots, \Phi(\vec{a}_\lambda'''(t))\};$
  replace: $P(t+1) := r_{\Theta_r}(P'''(t) \cup Q);$
  <communication step>
  $t := t + 1;$
end while

- **Decentralized GAs: cGA and dGA**



Repulsion Algorithm

- RA considers the words as equally charged particles
- RA calculates the repulsion forces among the particles
- RA moves a particle according to the resultant force exerted over it



$$\mathbf{f}_{ij} = \frac{1}{d_{ij}} \frac{\mathbf{p}_i - \mathbf{p}_j}{\sqrt{d_{ij}}}$$

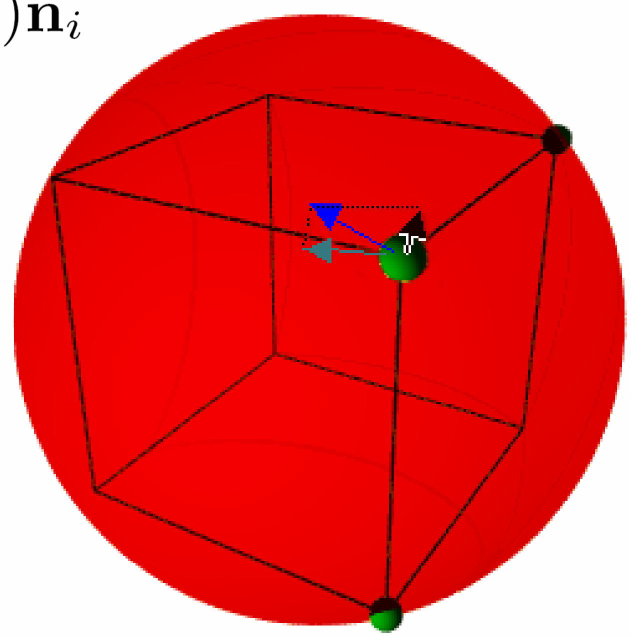
$$\mathbf{F}_i = \sum_{j=1, j \neq i}^{M} \mathbf{f}_{ij}$$

Repulsion Algorithm

- RA considers the words as equally charged particles
- RA calculates the repulsion forces among the particles
- RA moves a particle according to the resultant force exerted over it

$$\mathbf{F}_i^n = (\mathbf{F}_i \cdot \hat{\mathbf{n}}_i)\hat{\mathbf{n}}_i$$

$$\mathbf{F}_i^t = \mathbf{F}_i - \mathbf{F}_i^n$$

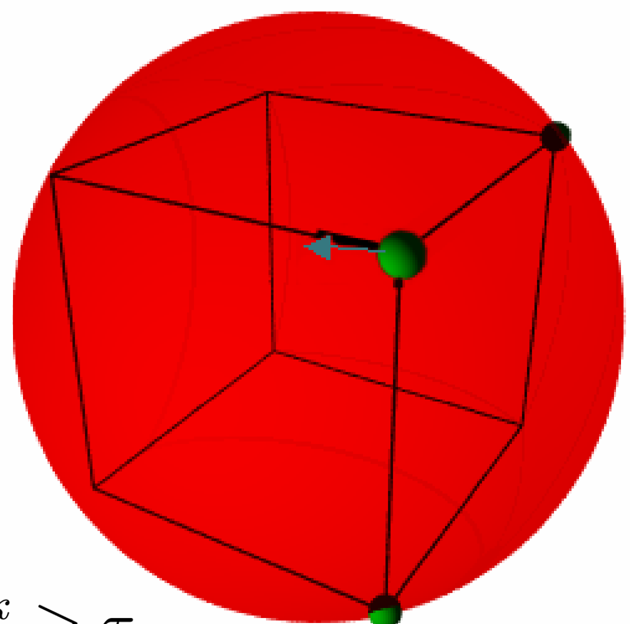


$$\mathbf{F}_i = \sum_{j=1, j \neq i}^{M} \mathbf{f}_{ij}$$

Repulsion Algorithm

- RA considers the words as equally charged particles
- RA calculates the repulsion forces among the particles
- RA moves a particle according to the resultant force exerted over it

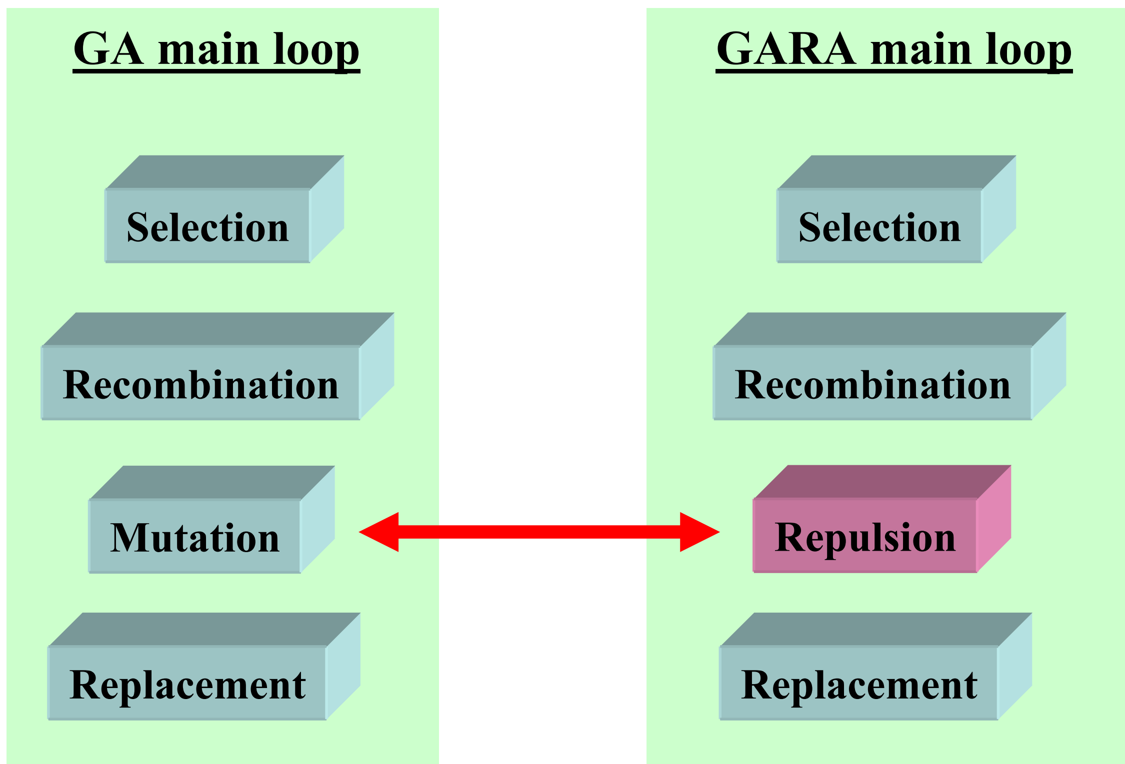
$$\mathbf{F}_i^t = \mathbf{F}_i - \mathbf{F}_i^n$$



$$m_i^k = \mathbf{F}_i^t \cdot \mathbf{e}_i^k \geq \tau$$

Hybrid Algorithms

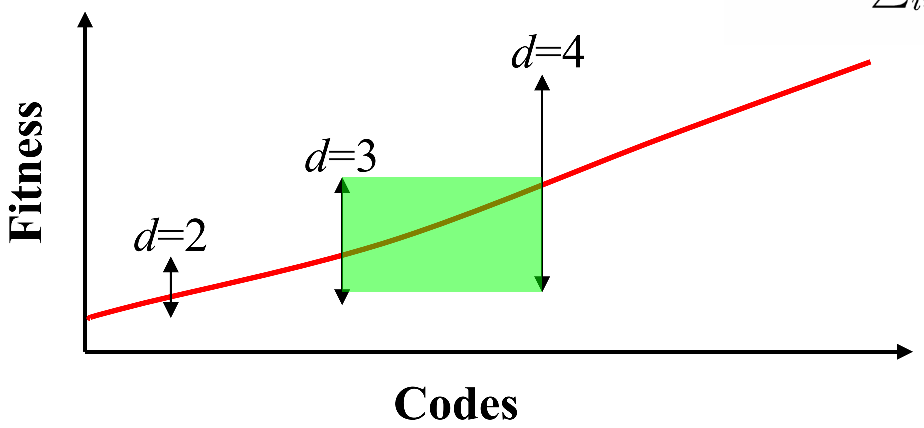
- **Hybridization: Inclusion of problem knowledge into the algorithm**
- **Two posible classes of hybrids:**
    - **Strong: Specific representation and operators**
    - **Weak: Combination of several algorithms (cooperation)**



Fitness

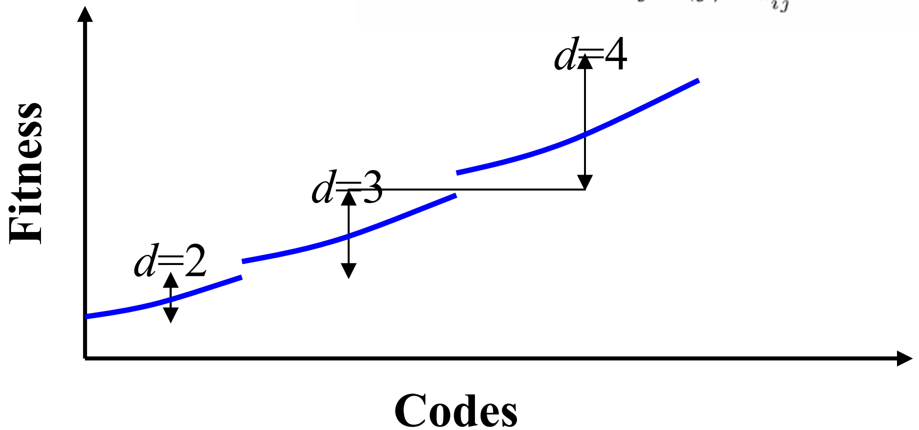
- **De Jong's function (possibly problematic):**

$$f(\mathbf{x}) = \frac{1}{\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\frac{1}{d_{ij}^2}}$$



- **Corrected function**

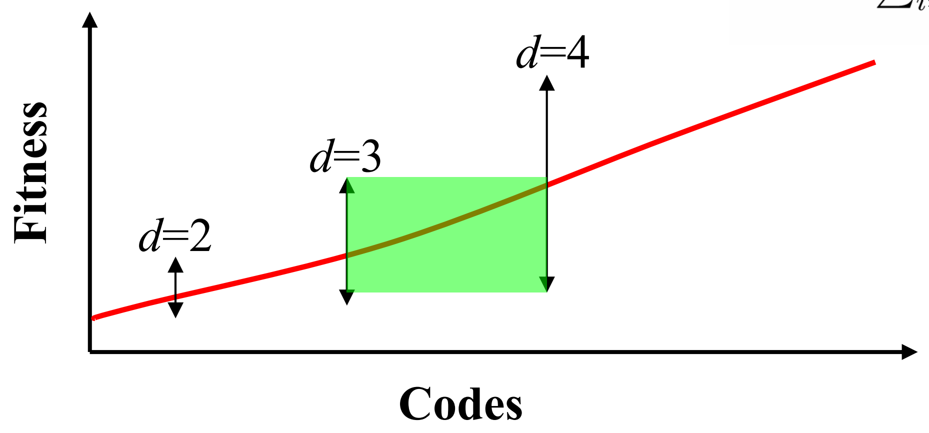
$$f(\mathbf{x}) = \frac{1}{\sum_{i=1}^{M}\sum_{j=1,j\neq i}^{M}\frac{1}{d_{ij}^2}} + \left(\frac{d_{min}}{12} - \frac{d_{min}^2}{4} + \frac{d_{min}^3}{6}\right)$$



Fitness

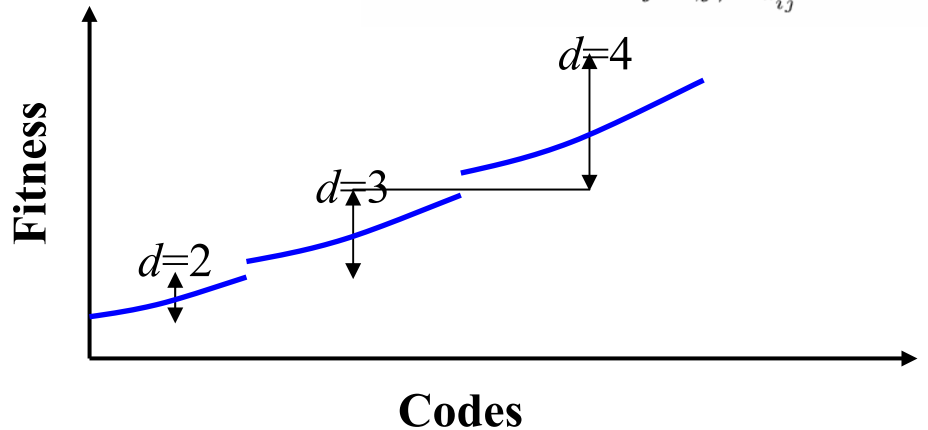
- **De Jong's function (possibly problematic):**

$$f(\mathbf{x}) = \frac{1}{\sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} \frac{1}{d_{ij}^2}}$$



- **Corrected function**

$$f(\mathbf{x}) = \frac{1}{\sum_{i=1}^{M} \sum_{j=1, j\neq i}^{M} \frac{1}{d_{ij}^2}} + \left(\frac{d_{min}}{12} - \frac{d_{min}^2}{4} + \frac{d_{min}^3}{6} \right)$$

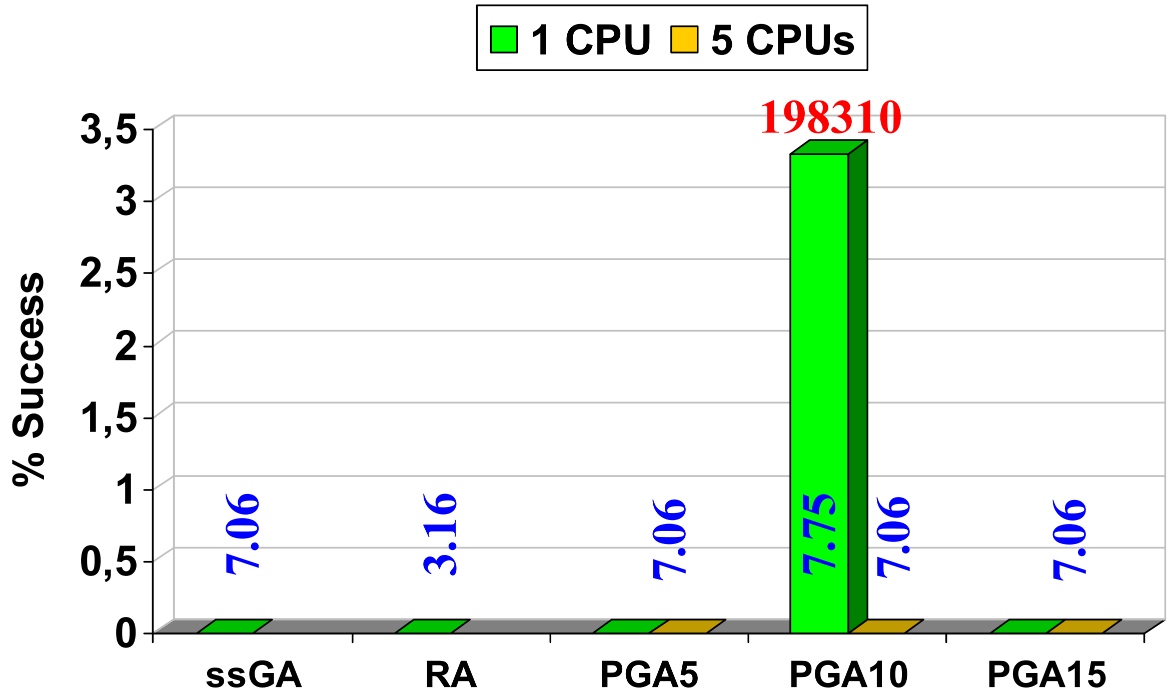


Parameters

- Genotype: concatenation of the codewords
- The same number of evaluations for all the algorithms ($2 \cdot 10^5$)
- Three distributed GAs: 5, 10 and 15 islands (same pop. size)
- Unidirectional ring for dGAs
- RA with $\tau = 0.001$
- Two classes of hybrids:
    - With recombination: PGARA*n*
    - Without recombination: PGRA*n*

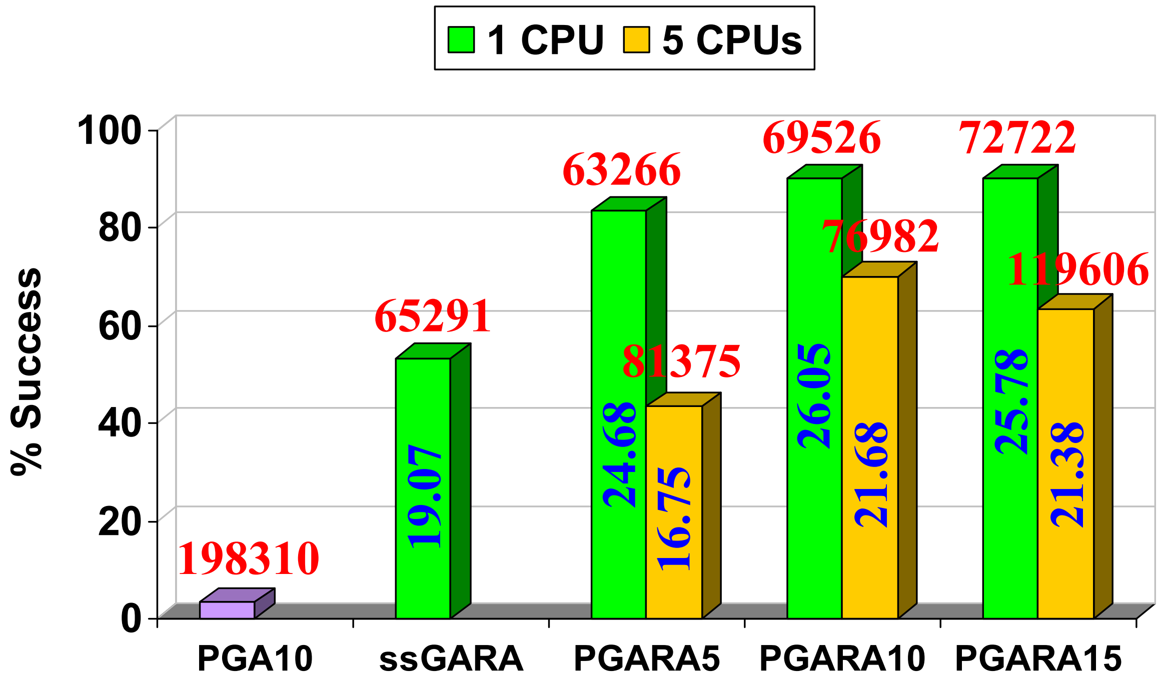
Results

- The pure algorithms obtain a very small success rate (<4%)
- The fitness of solutions given by RA are lower than GAs ones



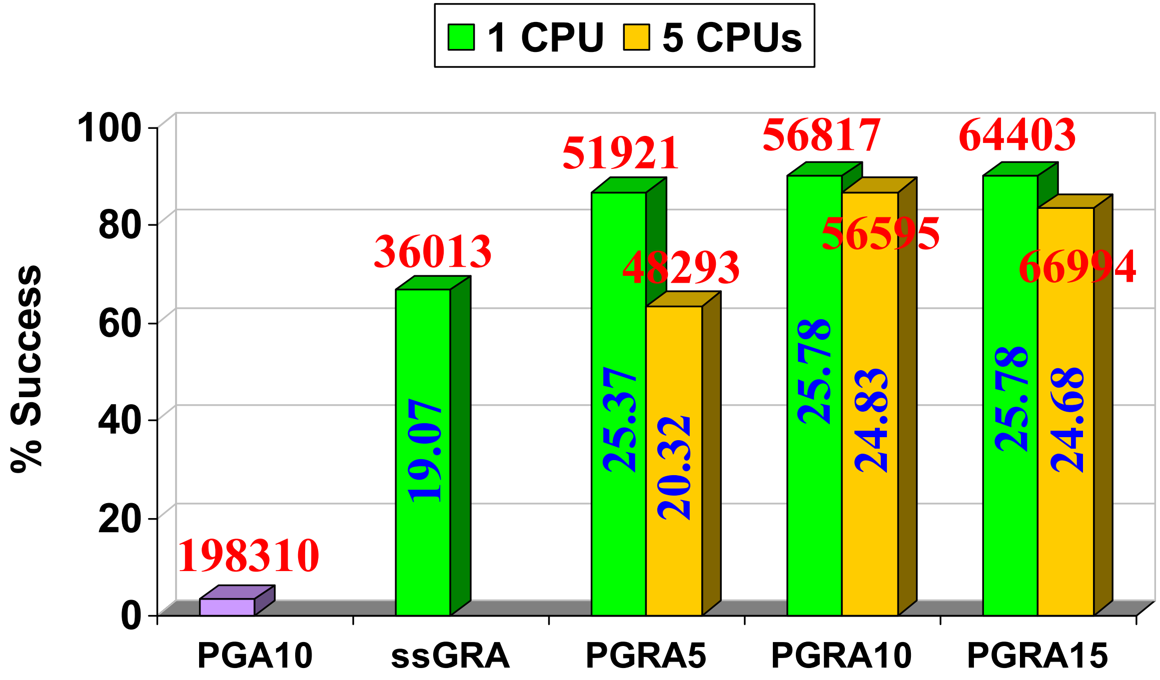
Results

- High success rate with hybrid algorithms
- Decentralized algorithms get still larger success than panmictic ones
- Results in 1 CPU better in both numerical effort and success



Results

- Higher hit rate in 5 CPUs
- Lower number of evaluations
- Improvement in the panmictic algorithm



Conclusions & Future Work

Conclusions

- Decentralization and hybridization with RA leverage the quality
- Increasing the number of islands do not always improves the search
- Results in 1 CPU better than 5 CPUs
- The recombination operator ruins the RA work

Future Work

- To study other algorithms for the problem (idea: scatter search + RA)
- To solve larger instances
- To extend the repulsion algorithm to other problems (Thomson)

THE END

Introduction

ECC Problem

Algorithms

Experiments

Conclusions & Future Work

Thanks for your attention !!!