A Hybrid Grouping Genetic Algorithm for Bin Packing

Emanuel Falkenauer

Simon Bartels, Matteo Pallocca, David Reijnders

Information and Computing Science Institute
Utrecht University,
The Netherlands

January 23, 2012

Universiteit Utrecht

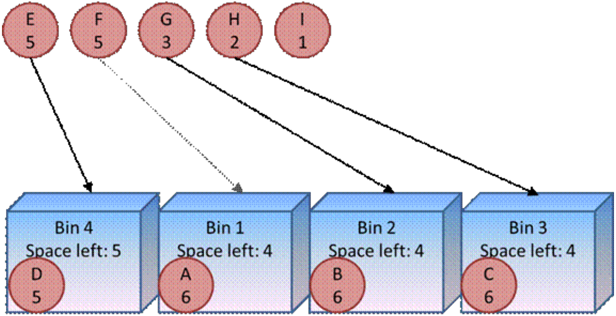
Motivation

- moving goods of certain weight with least lorries as possible
- assort commercial clips to advertisement breaks
- efficiently cutting cables/pipes of standardized length according to demand

Universiteit Utrecht

Problem Definition (informal)

- number of items of certain weight
- infinite amount of boxes/bins
    - have a maximum weight to hold
- trying to fit ALL items in as LEAST as possible bins



Problem Definition (one-dimensional)

given

O a finite multi-set of item sizes

C a constant describing bin's capacity

N a number of bin's

Is it possible to find a partitioning of O into N partitions such that the sum of elements in each partition does not exceed C?

$$P_j \subseteq O, \qquad j = 1, \ldots, N$$

1. $\cup_{j=1,\ldots,N} P_j = O$
2. $\forall j, k : j \neq k \Rightarrow P_j \cap P_k = \emptyset$
3. $\forall j : \sum_{o \in P_j} o \leq C$

Interest is on corresponding optimization problem.

Naive Approach

Instance example:

1 2 3 4 5 6
A D B C E B

Groups interpretation:
A={1} B={3,6} C={4} D={2} E={5}

Problems:

- High (exponential) degree of redundancy
- Larger research space

Crossover

Note that ABCADD == CADCBB

Standard two-point crossover:

A | B C | A D D

C | A D | C B B

would yield

C B C C B B

Problems:

- Two identical parents, different solutions
- Schema disruption

Mutation

A B D B A C → A B D E A C

A A A B B B → A A C B B B

Problems

- High chance to destroy a good solution
- The new solution is gonna be discarded soon

Falkenauer's Approach

New encoding structure

item part: group part
ADBCEB: BECDA

Strong points

- Order/Names of the groups is irrilevant to the GGA
- Genetic operators work only on group parts

Mutation

Choose between:

- Create new group
- Eliminate existing group
- Shuffle a small number of items among groups

Universiteit Utrecht

Inversion operator

A D B C E B: B E C D A
$\rightarrow$
A D B C E B: C E B D A

Aim:

- Shield good groups from destruction
- Make the proliferation of good schematas easier

Universiteit Utrecht

Crossover

$p1 = itemsP1 : ABCDEF$
$p2 = itemsP2 : abcd$

Algorithm:

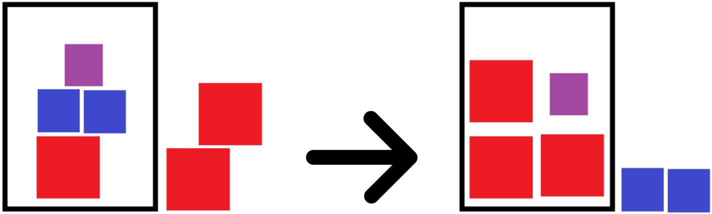
- Random selection of two crossing sites
- Content injection
- Eliminate duplicates from the 'old memberships'
- Apply previous points with inverted roles.

Local Search

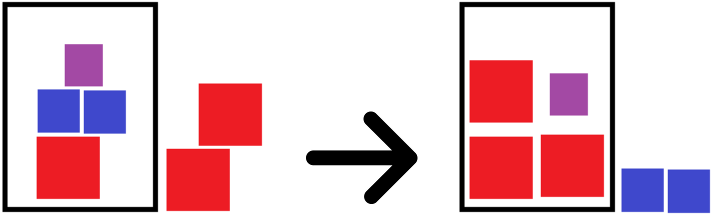
- reason why algorithm is called hybrid
- two operators
    - Martello and Toth
    - First Fit
- Silvano Martello and Paolo Toth *Lower Bounds and Reduction Procedures for the Bin Packing Problem*, in Discrete Applied Mathematics, vol. 22, North-Holland, Elsevier Science Publishers B.V., pp.59-70.

Local Search

- look at each bin
- try to **replace** 1,2 or 3 items
- with 1 or 2 unassigned items
- such that bin is filled better



Local Search



- bin is filled better
- easier to find bin for unassigned items

Universiteit Utrecht

Falkenauer's Fitness Function

- naive fitness function (number of bins) has large plateaus
- $f(s) := \dfrac{\sum_{j=1}^{n} (F_j / C)^k}{n}$

s	a solution
k	a constant ≥ 1
F_j	sum of item weights in bin j
C	bin capacity
n	the number of used bins

Falkenauer's Fitness Function

$$f(s) := \frac{\sum_{j=1}^{n} (F_j/C)^k}{n}$$

- larger k promotes fuller bins
- larger k punishes less filled bins
- perfect solution (all bins full)
  - $f(s) = 1$
- bad solution (bins nearly empty)
  - $f(s) \to 0$

Falkenauer's Fitness Function

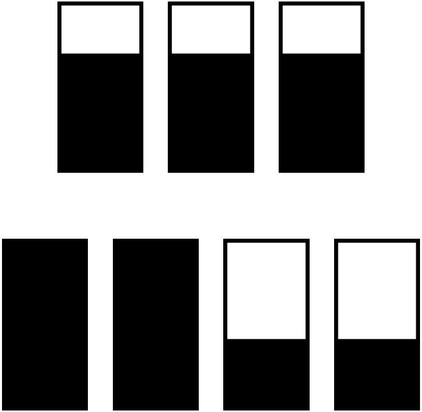
$$f(s) := \frac{\sum_{j=1}^{n} (F_j / C)^k}{n}$$

vs.

$$g(s) := n$$

Falkenauer's Fitness Function

$$f(s) := \frac{\sum_{j=1}^{n} (F_j/C)^k}{n} \qquad \text{vs.} \qquad g(s) := n$$



Falkenauer's Fitness Function

$$f(s) := \frac{\sum_{j=1}^{n} (F_j/C)^k}{n} \qquad \text{vs.} \quad g(s) := n$$

- $k \leq 2$ no problem
- according to Falkenauer
- no proof

Experiment Setup

- HGGA versus MTP (Martello and Toth Procedure)
- MTP:
  - Best method yet
  - Enumerative / Branch-and-bound heuristic
- HGGA:
  - 100 individuals, tournament size 2
  - 50 crossover, 33 mutation, 25 inversion
- First population: First Fit

Experimental Setup

- Two experiments designed
  - First: MTP's turf
  - Second: Most difficult BPP
- Shared BPP instances
- Notable: C++ versus FORTRAN on two similar PC's
- Computation limit

Experiment 1: Uniform Item Size Distribution

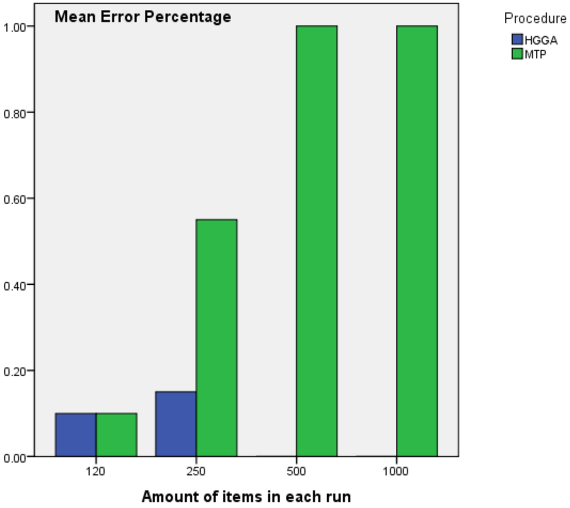
- Taken over from Martello and Toth [1990]
- MTP's most difficult setting
  - bin capacity: 150
  - item range: between 20 and 100
- Number of items: 120, 250, 500, 1000
- 20 instances each, 80 total

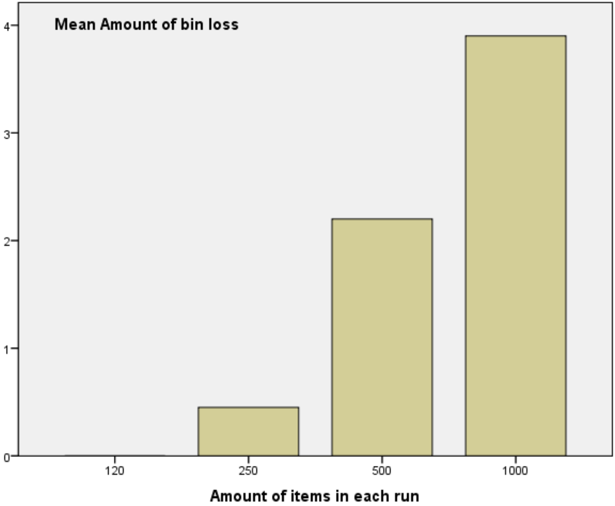
Universiteit Utrecht

Run	Theo	HGGA			MTP				
		Bins	Evals	Time	Bins	Loss	Loss%	Backs	Time
1	48	48	201	15.2	48	0	0.0	56	0.1
2	49	49	0	0.0	49	0	0.0	0	0.1
3	46	46	67	5.8	46	0	0.0	124935	29.0
4	49	49	804	50.4	49	0	0.0	74	0.0
5	50	50	0	0.0	50	0	0.0	0	0.0
6	48	48	268	19.4	48	0	0.0	43	0.1
7	48	48	268	19.0	48	0	0.0	69	0.0
8	49	49	335	21.7	49	0	0.0	54	0.0
9	**50**	**51**	134000	3668.7	51	0	0.0	10000000	3681.4
10	46	46	603	39.5	46	0	0.0	103	0.1
11	52	52	0	0.0	52	0	0.0	0	0.1
12	49	49	335	23.7	49	0	0.0	64	0.1
13	48	48	402	25.7	48	0	0.0	88	0.0
14	49	49	0	0.0	49	0	0.0	0	0.0
15	50	50	0	0.0	50	0	0.0	0	0.0
16	48	48	134	11.1	48	0	0.0	36	0.1
17	52	52	0	0.0	52	0	0.0	0	0.0
18	52	52	1340	76.1	52	0	0.0	48	0.0
19	49	49	201	14.3	49	0	0.0	24	0.0
20	**49**	**50**	134000	3634.7	50	0	0.0	7500000	3679.4
Averages				*381*		*0*	***0.0***		*370*
				6min					***6min***

Table 1 Uniform, 120 items.

Universiteit Utrecht


Mean Error Percentage
1.00
0.80
0.60
0.40
0.20
0.00
120
250
500
1000
Amount of items in each run
Procedure
HGGA
MTP

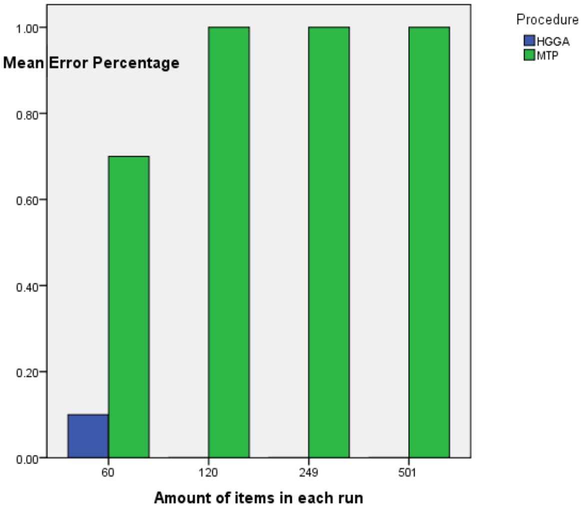

Mean Amount of bin loss
4
3
2
1
0
120
250
500
1000
Amount of items in each run

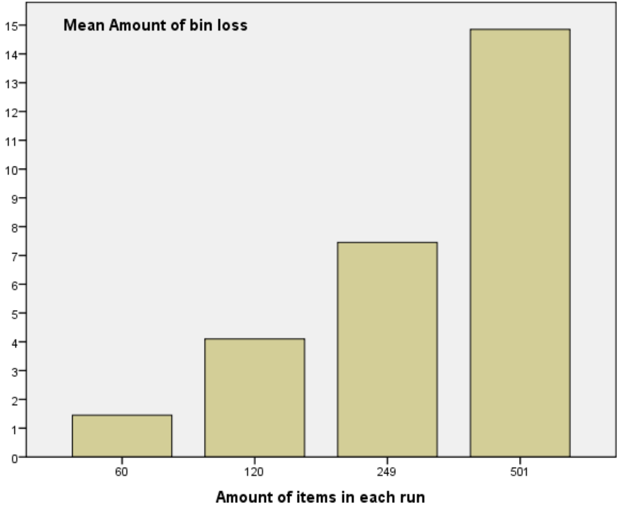
Experiment 2: Triplets

- Practical limits of HGGA
- Triplets
  - bin capacity: 1
  - item range: between 0.25 and 0.5
- Optimal: (0.4, 0.3, 0.3)
- Suboptimal: (0.4, 0.4) or (0.3, 0.3, 0.3)

Experiment 2: Triplets

- The experiment:construct known optima
  - bin capacity: 1000
  - One big between 380 and 490
  - Two small (with minimum 250) complete the bin
- Number of items: 60, 120, 249, 501
- 20 instances each, 80 total


Mean Error Percentage
1.00
0.80
0.60
0.40
0.20
0.00
60
120
249
501
Amount of items in each run
Procedure
HGGA
MTP


Mean Amount of bin loss
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
60
120
249
501
Amount of items in each run

Experiment: Conclusion

- HGGA performs better than MTP
- What if:
  - optimal solution != theoretical bound

Summary

- genetic algorithm for bin packing
  - encoding
  - crossover, mutation, inversion
  - local search operator
  - fitness function
- applicable to other grouping problems (e.g. graph colouring)

Critique

- careful with fitness function
  - behaviour for larger k unknown
- experiments are not as significant as they seem
  - MTP is faster in easier scenarios
  - stops automatically after finding global optima
- which algorithm to use depends on what you want