POPMUSIC : PARTIAL OPTIMIZATION METAHEURISTIC UNDER SPECIAL INTENSIFICATION CONDITIONS

Éric Taillard

EIVD

University of Applied Sciences of Western Switzerland

Yverdon-les-bains, Switzerland

CONTENT OF THE TALK

Popmusic : general idea, frame, choices

Related concepts

Applications

- VRP
- Turbine runner balancing
- Clustering
- Cartographic labelling

Conclusions

POPMUSIC

General idea

Start from an initial solution

Decompose solution into parts

Optimize a portion (several parts) of the solution

Repeat until the optimized portions cover the entire solution

POPMUSIC FRAME

Solution $S = s_1 \cup s_2 \cup \ldots \cup s_p$ // p disjoint parts

$O = \varnothing$ // Set of " optimized " seed parts

While $O \neq S$, repeat // Improving method

1. Choose a seed part $s_i \notin O$
2. Create a sub-problem R composed of the r " closest " parts $\in S$ from s_i // r : parameter
3. Optimize sub-problem R
4. If R improved : set $O \leftarrow O \setminus R$

   Else : set $O \leftarrow O \cup s_i$

POPMUSIC CHOICES

Definition of a part

Distance between two parts

Parameter *r*

Optimization procedure

Variants :

slower :	set $O \leftarrow \emptyset$	instead of	set $O \leftarrow O \setminus R$
faster :	set $O \leftarrow O \cup R$	instead of	set $O \leftarrow O \cup s_i$

RELATED CONCEPTS

Candidate list, strongly determined and consistent variables (Glover)

“Chunking” (Woodruff)

Large neighbourhoods (Shaw)

VDNS (Hansen & Mladenovic)

Decomposition methods

POPMUSIC FOR VRP (TAILLARD 1993, ...)

Part:

Vehicle tour

Distance between parts:

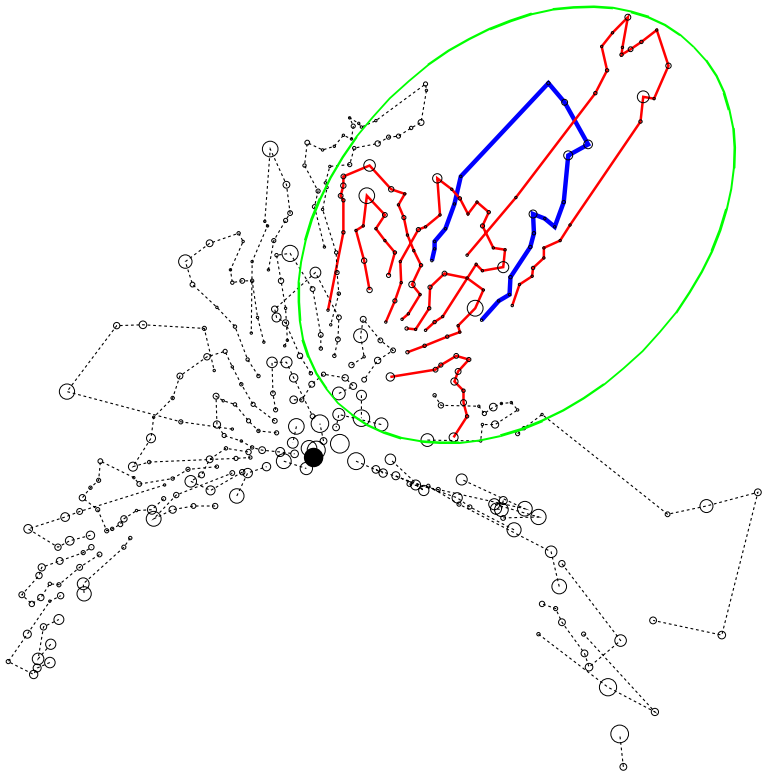
Polar distance between centres of gravity

Optimization process:

Basic taboo search

Particularity:

Many simultaneous optimization processes, treating all tours at each iterations



OTHER APPLICATIONS FOR VRP

Rochat & Semet 1994

Particularity: Repeat POPMUSIC with increased parameter *r*

First VNS application ?

Shaw 1998 (Large Neighbourhood)

Part: Customer

Distance: Euclidean distance + random component

Optimization process: Optimum or heuristic re-insertion (with constraint logic programming)

POPMUSIC FOR BALANCING PROBLEMS

Turbine runner balancing :

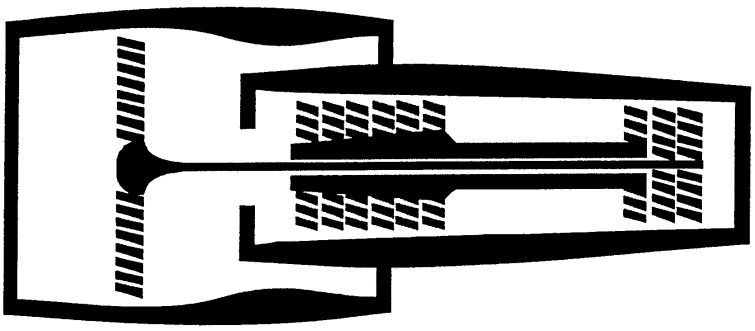
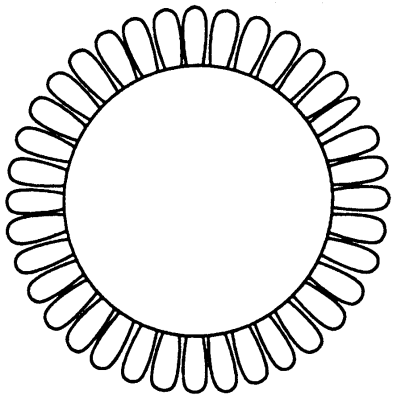


Fig. 1. Schematic figure of a jet engine.



Source : Mason & Rönnqvist, C&OR 24, 1997

n blades of weight w_i (i = 1, …, n)

n angular positions $\theta_i = i/2\pi$ (i = 1, …, n) or, more generally : cartesian coordinates (x_i, y_i)

Objective : find a positions p_i (i = 1, …, n) for each blade that minimizes : $\left(\sum_{i=1}^{n} w_i \cdot x_{p_i} \right)^2 + \left(\sum_{i=1}^{n} w_i \cdot y_{p_i} \right)^2$

TURBINE BALANCING

Alternate formulation (Laporte & Mercure, EJOR 35, 1988) :

Quadratic assignment problem with :

flows matrix $f_{ij} = w_i w_j$

distances matrix $d_{ij} = cos(\theta_i - \theta_j)$

Objective : find a permutation p that minimizes : $\sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} \cdot d_{p_i p_j}$

Less general

Works only for angular positions

Container vessel loading ?

More complex

Objective computation $O(n^2)$ versus initial formulation $O(n)$

POPMUSIC CHOICES

Part:

Mechanical part

Distance:

Weight difference

Optimization process:

Basic taboo search (transposition neighbourhood, Taillard 1991)

NUMERICAL RESULTS (BASIC TABOO SEARCH)

Comparison with Reverse Elimination Method and Star Shape Diversification (Sondergeld & Voß 1996)

Turbines with 5..80 blades	REM/1	REM/10	SSA1/1	SSA1/10	SSA2/1	SSA2/10	Basic taboo search
$\sum$ excentricity	37.28685	37.55715	37.32012	37.47627	37.30784	37.30163	37.26166
$\sum$ CPU time [s]	> 5000	> 5000	> 5000	> 5000	> 11000	> 11000	109

NUMERICAL RESULTS

Number of blades	$100\left(\frac{\text{Basic taboo(10000)}}{\text{POPM(22, 1000)}} - 1\right)$	CPU POPM. [s. Sparc5]
30	7.2	3.5
40	40.0	7.2
50	52.1	12.8
60	49.4	18.0
70	76.0	21.4
80	65.2	26.5
100	23.3	37.2
100	56.0	55.5
100	35.3	65.1
100	56.6	49.3
100	11.5	58.5
100	32.0	46.8
100	77.7	57.9
100	47.1	51.3
100	51.3	62.5
100	26.2	45.1

POPMUSIC FOR CLUSTERING

Part :

Elements belonging to a cluster

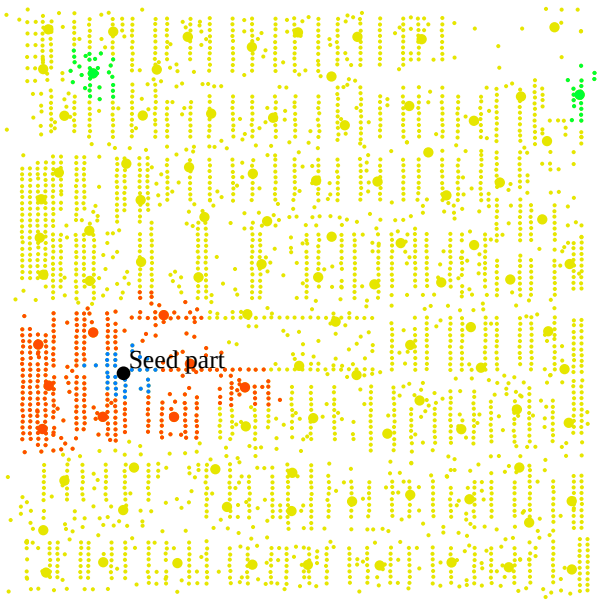
Distance :

Average dissimilarity between elements of different groups,

Distance between centres

Optimization process :

Improving method based on candidate list, relocation of a centre, stabilization (CLS)



NUMERICAL RESULTS

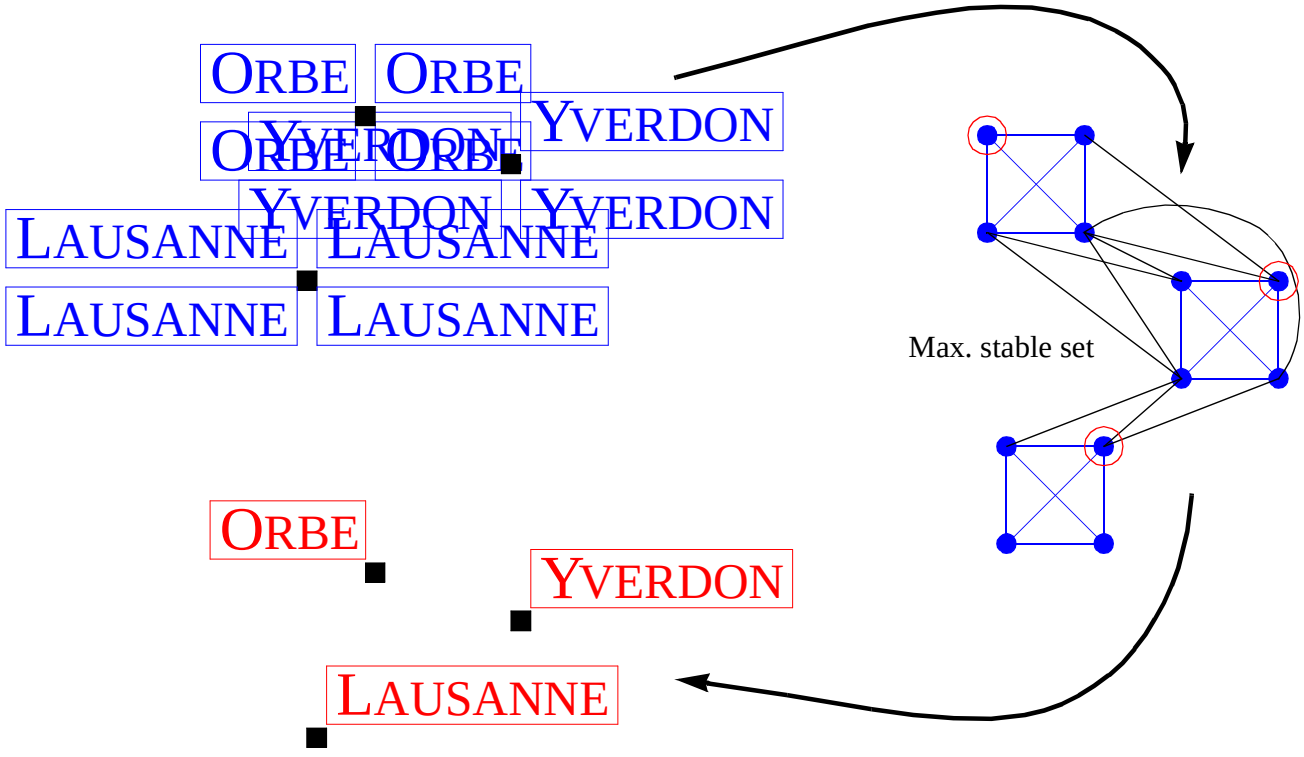
Minimum sum of squares

Problem instance : TSPLIB pcb3038.

Results of RVNS and VDNS from Hansen & Mladenovic 1999.

Problem		Quality [% above best known]				Computational time [s SPARC10]			
p	Best solution known	RVNS	VDNS	POPM. (6, 40)	POPM. (10, 100)	RVNS	VDNS	POPM. (6, 40)	POPM. (10, 100)
100	47685934.0	2.34	0.73	1.19	0.44	153	1132	145	505
150	30524769.8	3.13	1.44	1.16	0.58	153	1676	111	355
200	21875113.9	2.49	1.10	1.07	0.50	160	2124	96	262
250	16621446.4	2.56	1.34	1.35	0.76	182	2954	89	234
300	13289633.4	2.50	1.57	1.58	0.78	229	3151	82	205
350	11019171.4	2.60	1.36	1.69	0.76	231	3760	75	179
400	9362179.2	3.35	1.82	1.40	0.66	165	3446	72	170
450	8101618.7	3.47	1.71	1.61	0.80	242	4152	69	163
500	7102678.4	2.85	1.86	1.70	0.90	204	4060	68	156

CARTOGRAPHIC LABEL PLACEMENT



POPMUSIC CHOICES

Part:

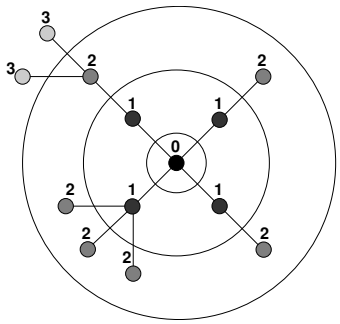
Object to label

Distance between parts:

Minimum number of edges needed to connect parts

Vertex ≡ object

Edge ∃ possible conflict in labelling the objects associated to vertices connected



Optimization process:

Tuned taboo search (Yamamoto, Camara, Nogueira Lorena, 2002)

NUMERICAL RESULTS

% of good placements					
Method	Problem size				
	100	250	500	750	1000
POPMUSIC	100	100	99.6	97.4	92.3
POPMUSIC fast	100	100	99.5	97.2	91.6
CGA (best) (Yamamoto, Nogueira Lorena, 2003)	100	100	99.6	97.1	90.7
CGA(av.) (Yamamoto, Nogueira Lorena, 2003)	100	100	99.6	96.8	90.4
Tabu (Yamamoto, Camara, Nogueira Lorena, 2002)	100	100	99.2	96.8	90.00
GA with masking (Verner, Wainwritht, Schönenfeld, 1997)	100	99.98	98.79	95.99	88.96
GA (Verner, Wainwritht, Schönenfeld, 1997)	100	98.40	92.59	82.38	65.70
Simulated Annealing (from Christensen et al. 1995)	100	99.90	98.30	92.30	82.09
Zoraster(from Christensen et al. 1995)	100	99.79	96.21	79.78	53.06
Hirsh (from Christensen et al. 1995)	100	99.58	95.70	82.04	60.24
3-Opt Gradient Descent (from Christensen et al. 1995)	100	99.76	97.34	89.44	77.83
2-Opt Gradient Descent (from Christensen et al. 1995)	100	99.36	95.62	85.60	73.37
Gradient Descent (from Christensen et al. 1995)	98.64	95.47	86.46	72.40	58.29
Greedy (from Christensen et al. 1995)	95.12	88.82	75.15	58.57	43.41
CPU time (Pentium III, 745MHz, ?)					
POPMUSIC	0.0	0.0	0.3	3.5	20
POPMUSIC2 fast	0.0	0.0	0.2	1.3	4
CGA (best) (Yamamoto, Nogueira Lorena, 2003)	0	0.6	21.5	228	1227
CGA(av.) (Yamamoto, Nogueira Lorena, 2003)	0	0.6	21.5	196	982
Tabu (Yamamoto, Camara, Nogueira Lorena, 2002)	0	0	1.3	76	353

CONCLUSIONS

Easy to implement

Simple frame

Basic improvement method (no candidate list, intensification, diversification)

Can be used with an exact method

Low complexity

Grows typically linearly with problem size

Future

Better implementations for *p*-median, map labelling, VRP

Projects and collaborations with other teams ?

ASRO competition ?