

Παρακάτω παρατηρούμε τα εξής αποτελέσματα από τους αλγορίθμους:

Images with local search MIN:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	threshold	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	2	100	10	1	16345194	15101118	0.216542	0.20006	5
euro-night-0000100.in stance	convex hull	2	100	10	1	23110406	22522072	0.306168	0.298374	44
euro-night-0000400.in stance	Incremental (1a)	2	400	10	1	16849010	16648076	0.206847	0.204381	33
euro-night-0000400.in stance	convex hull	2	400	10	1	28885876	28868948	0.354618	0.354411	7080
uniform-0000400-1.in stance	convex hull	2	400	10	1	196196888	192704622	0.356837	0.350486	8276

Images with local search MAX:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	threshold	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	3	100	10	1	57276158	58464406	0.758798	0.77454	7
euro-night-0000100.in stance	convex hull	3	100	10	1	66991396	67185486	0.887506	0.890077	31
euro-night-0000400.in stance	Incremental (1a)	3	400	10	1	59947742	60593586	0.73595	0.743879	67
euro-night-0000400.in stance	convex hull	3	400	10	1	70842394	70842394	0.869699	0.869699	4793
uniform-0000400-1.in stance	convex hull	3	400	10	1	427996102	428768852	0.778427	0.779833	8166

Images with simulated annealing MIN with Local Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	48839118	44824274	0.647023	0.593834	28
euro-night-0000100.in stance	convex hull	1	100	1000	44996748	43504360	0.596119	0.576348	64
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	41443050	40153250	0.508777	0.492943	111
euro-night-0000400.in stance	convex hull	1	400	1000	43391204	43694304	0.532694	0.536415	8667
uniform-0000400-1.in stance	convex hull	1	400	1000	287883524	290790230	0.523594	0.528881	6168

Images with **simulated annealing MAX** with Local Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	38796070	40013350	0.513972	0.530099	27
euro-night-0000100.in stance	convex hull	1	100	1000	48959238	47747374	0.648615	0.63256	80
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	37137170	38008834	0.455916	0.466617	136
euro-night-0000400.in stance	convex hull	1	400	1000	43561076	44180622	0.534779	0.542385	7374
uniform-0000400-1.in stance	convex hull	1	400	1000	19451120	19276788	0.583879	0.578646	97
euro-night-0001000.in stance	convex hull	1	1000	1000	44202640	42678524	0.51256	0.494887	301480

Images with simulated annealing MIN with Global Step:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000100.in stance	Incremental (1a)	1	100	1000	38714994	38638130	0.512898	0.51188	59
euro-night-0000100.in stance	convex hull	1	100	1000	45190108	39699860	0.598681	0.525946	104
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	37186608	36763320	0.456523	0.451326	441
euro-night-0000400.in stance	convex hull	1	400	1000	40739210	36380316	0.500136	0.446624	6162
uniform-0000400-1.in stance	convex hull	1	400	1000	309739078	287703700	0.563345	0.523267	6530
uniform-0000400-1.in stance	Incremental(1a)	1	400	1000	296735046	263127874	0.539693	0.47857	477

Images with simulated annealing MAX with Global Step:

Name File	greedy Algorithm (sort)	Edge Selection	Points	L	old Area	new Area	old Ratio	new Ratio	Time(ms)	
euro-night-0000100.instance	Incremental (1a)	3	100	1000	40774882	38342942	0.540188	0.507969	4	
euro-night-0000100.instance	convex hull	3	100	1000	38249430	45902142	0.506731	0.608114	53	
euro-night-0000400.instance	Incremental (1a)	3	400	1000	40081254	40455528	0.492059	0.496654	29	
euro-night-0000400.instance	convex hull	3	400	1000	47406546	48550198	0.581988	0.596028	11988	
uniform-0000400-1.instance	convex hull	3	400	1000	16961794	19025528	0.509155	0.571104	58	
euro-night-00005000.instance	Incremental (1a)	1	5000	1000	45412384	45412384	0.526588	0.526588	263414	

Images with simulated annealing MIN with Sub Division:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	M	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	70	38122646	36315150	0.468014	0.445824	129
euro-night-0000400.in stance	convex hull	1	400	1000	70	40088738	39592142	0.492151	0.486054	221
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	100	38916564	37621510	0.47776	0.461862	101
euro-night-0000400.in stance	convex hull	1	400	1000	100	40551166	39102260	0.497828	0.48004	297
uniform-0000400-1.in stance	convex hull	1	400	1000	100	275958398	276307262	0.501905	0.50254	258
euro-night-00005000.i nstance	Incremental (1a)	1	5000	1000	100	144911496	146125072	0.411771	0.41522	2534

Images with **simulated annealing MAX** with Sub Division:

Name File	greedy Algorithm(sort)	Edge Selection	Points	L	M	old Area	new Area	old Ratio	new Ratio	Time(ms)
euro-night-0000400.in stance	Incremental (1a)	1	100	1000	70	38235628	37431204	0.506548	0.495891	32
euro-night-0000400.in stance	convex hull	1	100	1000	70	41067664	42000298	0.544067	0.556422	124
euro-night-0000400.in stance	Incremental (1a)	1	400	1000	100	36062792	34894034	0.442726	0.428378	138
euro-night-0000400.in stance	convex hull	1	400	1000	100	39240010	39698042	0.481731	0.487354	376
uniform-0000400-1.in stance	convex hull	1	400	1000	100	17301378	17548396	0.519349	0.526764	109
euro-night-0005000.in stance	convex hull	1	400	1000	100	38485906	37761028	0.446271	0.437865	867

Παρατηρήσεις:

- Τα αρχεία uniform είναι πιο αργά από τα euronight.
- Ο local search βελτιώνει πάντα το εμβαδό ενώ ο simulated annealing όχι πάντα.
- Όλοι οι αλγόριθμοι είναι αρκετά γρήγοροι, στον global step βέβαια έχουμε βάλει upper bound στα iterations ίσο με $2 \cdot L$.
- Κάποιες φορές ο greedy αλγόριθμος βρίσκει αρκετά καλό πολύγωνο με αποτέλεσμα ο optimization αλγόριθμος να το κάνει λίγο χειρότερο καθώς του λέμε να κάνει κάποια κίνηση.
- Επειδή υπάρχει σε πολλά σημεία των αλγορίθμων randomization για να αποφύγουμε την εξαντλητική αναζήτηση σε κάθε βήμα, είναι αναμενόμενο να μην βρίσκουμε σε κάθε run την βέλτιστη λύση.
- Όταν χρησιμοποιούμε τον incremental ως greedy αλγόριθμο, το πρόγραμμα τερματίζει πιο γρήγορα.

- Παρατηρούμε ότι οι αλγόριθμοι που ψάχνουν το μαξ εμβαδόν μερικές φορές το πολύγωνο έχει καλύτερο εμβαδόν στην αρχή και όχι μετά από τις βελτιώσεις.